Atlas 300I 推理卡

用户指南（型号 3000）

文档版本 26
发布日期 2023-01-09

华为技术有限公司

版权所有 © 华为技术有限公司 2023。 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。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： https://e.huawei.com

前言

概述

本文档详细的描述了华为Atlas 300I 推理卡（型号 3000）的外观外形、产品规格、所需工具的安装和使用方法，以及日常管理等内容。

读者对象

本文档主要适用于以下人员：

- 企业管理员
- 企业终端用户

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
危险	表示如不避免则将会导致死亡或严重伤害的具有高等级风险的危害。
警告	表示如不避免则可能导致死亡或严重伤害的具有中等级风险的危害。
注意	表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。
须知	用于传递设备或环境安全警示信息。如不避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
说明	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

命令行格式约定

格式	意义
粗体	命令行关键字（命令中保持不变、必须照输的部分）采用**加粗**字体表示。
斜体	命令行参数（命令中必须由实际值进行替代的部分）采用*斜体*表示。
[ ]	表示用“[ ]”括起来的部分在命令配置时是可选的。
{ x | y | ... }	表示从两个或多个选项中选取一个。
[x | y | ...]	表示从两个或多个选项中选取一个或者不选。
{ x | y | ... }*	表示从两个或多个选项中选取多个，最少选取一个，最多选取所有选项。
[x | y | ...]*	表示从两个或多个选项中选取多个或者不选。
&<1-n>	表示符号&前面的参数可以重复1～n次。
#	由“#”开始的行表示为注释行。

修改记录

文档版本	发布日期	修改说明
26	2023-01-09	第二十六次正式发布。 ● 修改**2.3 系统框图**。 ● **4.1 基本规格**优化AI处理器、内存规格描述，尺寸增加宽度。 ● **4.2 环境条件**的海拔高度增加ASHRAE 2015标准说明。 ● 修改**4.5 热插拔**支持情况。 ● **4.6.1 散热要求**优化说明内容。 ● 新增**5 硬件兼容性**。 ● **11.2 带外管理**补充参考文档链接。
25	2022-07-29	第二十五次正式发布。 新增**7 安装与维护（适用于5.1.RC2及以上版本）**、**10.1.2 npu-smi命令介绍（适用于5.1.RC及以上版本）**。
24	2022-07-11	第二十四次正式发布。 修改软件包名称。

文档版本	发布日期	修改说明
23	2022-05-25	第二十三次正式发布。 修改**12.1 简介**。
22	2022-04-26	第二十二次正式发布。 新增**8.4.9 驱动卸载过程中出现进程占用报错**。 修改**12 容器内安装**。
21	2022-03-07	第二十一次正式发布。 **8.2.5 安装驱动**修改文档链接。 优化默认数据。 **8.2.5.3.1 安装依赖**增加Ubuntu 20.04操作系统安装依赖。 新增： ● **10.1.3.4.49 查询NPU设备首次上电的时间** ● **10.1.3.4.50 查询所有芯片的进程占用内存** ● **10.1.3.4.51 查询指定芯片的进程占用内存** 修改： ● **10.1.3.4.1 查询基本信息** ● **10.1.3.4.13 查询所有芯片统计信息** ● **10.1.3.4.14 查询指定芯片统计信息** ● **10.1.3.5.9 配置ECC使能状态**
20	2022-01-06	第二十次正式发布。 **8.2.5 安装驱动**和**8.2.7 卸载驱动**增加说明。
19	2021-11-01	第十九次正式发布。 新增： **8.5.3 相关工具**。 修改： **12.1 简介**章节刷新了兼容性关系列表。
18	2021-10-22	第十八次正式发布。 修改： **8.2.5.1 确认安装方式**。 **8.2.6.2 安装固件（适用于.rpm格式）**。
17	2021-08-24	第十七次正式发布。 新增**10.2 NPU一键式日志收集工具**。

文档版本	发布日期	修改说明
16	2021-07-31	第十六次正式发布。 调整大纲，新增**8 安装与维护（适用于1.0.11-1.0.15版本）**和**10.1.3 npu-smi命令介绍（适用于1.0.11-1.0.15版本）**。
15	2021-04-29	第十五次正式发布。 **9.5.3.1 安装依赖**章节增加SUSE12SP5、Linx 6.0.90操作系统安装依赖。
14	2021-03-30	第十四次正式发布。 **10.1.1 安装npu-smi工具**删除了《Atlas 300I 推理卡 1.0.0.SPC200以下 升级指导书（型号 3000, 3010）》引用。
13	2021-03-04	第十三次正式发布。 调整大纲，新增安装流程、确认操作系统内容。
12	2021-01-25	第十二次正式发布。 刷新了“安装依赖”。 新增： **9.10.10 安装驱动过程中出现“cp: cannot overwrite directory '/usr/local/sbin/npu-smi' with non-directory”的报错**。 **9.10.11 安装驱动或启动服务过程中出现“please use cmd (lsattr /etc/sudoers) to check permission!”报错**。 **9.10.12 Ascend 310芯片无法正常启动，系统日志出现如下打印信息：dev id 0, The address dont comply with the NVME protocol(must be aligned with 4KB).**。
11	2020-11-30	第十一次正式发布。 刷新了HwHiAiUser用户的使用版本。
10	2020-11-19	第十次正式发布。 刷新了软件包获取方式。

文档版本	发布日期	修改说明
09	2020-10-26	第九次正式发布。 调整大纲，新增NPU 20.1.X系列版本软件包的内容。 新增： ● **10.1.4.5.33 查询所有芯片的PCIE误码计数**。 ● **10.1.4.5.34 查询指定芯片的PCIE误码计数**。 ● **10.1.4.6.10 清除所有芯片的PCIE误码计数**。 ● **10.1.4.6.11 清除指定芯片的PCIE误码计数**。 ● **9.10.5 驱动重编译正常但安装上报modprobe失败**。 ● **9.10.6 Centos7.6系统宿主机目录挂载容器创建“/var/log/npu/slog/slogd”时，提示：Permission denied**。 ● **9.10.7 Centos7.6系统宿主机目录挂载容器拉起slogd时，提示：[WARNING] Init config path failed and continue**。 ● **9.10.8 Ascend_monitor通讯异常**。 ● **9.10.9 安装驱动和固件后重启，“/var/log/npu/slog/host-0”日志中产生“[ERROR] raise script failed, cmd run failed, errno = 0”报错**。 修改： ● 增加源码驱动包的检查项。 ● 增加源码驱动包安装相关步骤。 ● **10.1.4.6.6 复位标卡**章节增加通过带内复位标卡参数说明。 ● **10.1.4.6.7 复位芯片**章节增加通过带内复位芯片参数说明。
08	2020-09-14	第八次正式发布。 ● 新增覆盖安装过程中驱动和固件的安装顺序。 ● **10.1.4.6.6 复位标卡**章节增加使用限制说明。 ● **10.1.4.6.7 复位芯片**章节增加使用限制说明。

文档版本	发布日期	修改说明
07	2020-08-28	第七次正式发布。 修改： **10.1.4.6.9 设置AI芯片的IP信息**章节增加NPU 20.0.0以下版本配置IP必须先使能ssh功能的说明。 **12.1 简介**章节增加宿主机与容器操作系统兼容性关系列表。
06	2020-07-15	第六次正式发布。 调整大纲，新增NPU 20.0.0及以上软件包安装的内容。 新增以下章节： ● **10.1.4.5.32 查询AI芯片的IP信息**。 ● **10.1.4.6.9 设置AI芯片的IP信息**。 ● **12 容器内安装**。
05	2020-07-08	第五次正式发布。 修改： 操作系统内核版本信息。
04	2020-04-30	第四次正式发布。 修改： 软件包获取路径。
03	2020-03-30	第三次正式发布。 调整大纲，新增驱动安装与维护，工具使用的具体操作步骤。
02	2019-11-28	第二次正式发布。 增加参见文档的超链接。
01	2019-11-22	第一次正式发布。

目 录

前言 ii

1 安全 1

1.1 通用安全注意事项 1

1.2 设备上的标志 2

1.3 电气安全 3

2 产品简介 5

2.1 概述 5

2.2 外观 5

2.3 系统框图 6

3 产品特点 7

3.1 性能特点 7

3.2 可维护性特点 7

3.3 典型应用场景 7

4 产品规格 9

4.1 基本规格 9

4.2 环境条件 10

4.3 时钟要求 11

4.4 电源管理 11

4.5 热插拔 11

4.6 散热规格 11

4.6.1 散热要求 11

4.6.2 散热规格 12

4.6.3 过温保护 12

5 硬件兼容性 14

6 信号管脚 15

6.1 管脚定义 15

7 安装与维护（适用于 5.1.RC2 及以上版本） 22

8 安装与维护（适用于 1.0.11-1.0.15 版本） 23

8.1 用户必读 23

8.2 物理机安装 24

8.2.1 安装流程 25
8.2.2 确认操作系统 26
8.2.3 获取软件包 26
8.2.4 创建运行用户 27
8.2.5 安装驱动 28
8.2.5.1 确认安装方式 28
8.2.5.2 二进制文件直接安装 32
8.2.5.2.1 安装驱动（适用于.run 格式） 32
8.2.5.2.2 安装驱动（适用于.rpm 格式） 34
8.2.5.2.3 安装驱动（适用于.deb 格式） 35
8.2.5.3 源码编译安装 39
8.2.5.3.1 安装依赖 39
8.2.5.3.2 安装驱动（适用于.run 格式） 42
8.2.5.3.3 安装驱动（适用于.rpm 格式） 45
8.2.5.3.4 安装驱动（适用于.deb 格式） 47
8.2.5.4 重构驱动包安装 50
8.2.5.4.1 安装依赖 51
8.2.5.4.2 重构驱动包 51
8.2.5.4.3 安装驱动（适用于.run 格式） 53
8.2.6 安装固件 56
8.2.6.1 安装固件（适用于.run 格式） 56
8.2.6.2 安装固件（适用于.rpm 格式） 57
8.2.6.3 安装固件（适用于.deb 格式） 59
8.2.7 卸载驱动 60
8.2.7.1 确认卸载方式 61
8.2.7.2 卸载驱动（适用于.run 格式） 61
8.2.7.3 卸载驱动（适用于.rpm 格式） 62
8.2.7.4 卸载驱动（适用于.deb 格式） 63
8.2.8 卸载固件 64
8.2.8.1 卸载固件（适用于.run 格式） 64
8.2.8.2 卸载固件（适用于.rpm 格式） 65
8.2.8.3 卸载固件（适用于.deb 格式） 66
8.3 常用操作 66
8.3.1 通过 PuTTY 登录（网口方式） 66
8.3.2 使用 WinSCP 传输文件 68
8.3.3 设置用户有效期 70
8.3.4 配置 Host 侧 IP 地址通过 SSH 方式登录 Device 71
8.4 FAQ 73
8.4.1 升级系统内核版本时未卸载 run 包导致 run 包不可用 73
8.4.2 deb 包安装、升级或卸载失败后的处理方式 74
8.4.3 root 用户切换到普通用户需要输入密码权限 75
8.4.4 如何检查 device 是否正常运行 75

8.4.5 复位或下电导致 Device 无法正常启动.....76
8.4.6 软件包格式混用安装或升级恢复方法.....76
8.4.7 安装软件包时提示解压失败.....77
8.4.8 驱动加载超时，CPU 打印“soft lockup”报错.....77
8.4.9 驱动卸载过程中出现进程占用报错.....79
8.4.10 驱动安装过程中出现 dkms 编译失败报错.....80
8.4.11 安装 deb 包时出现“binary opetator expected”报错.....80
8.5 参考.....81
8.5.1 参数说明/常用命令.....81
8.5.2 驱动安装过程中所需的 Linux 工具.....85
8.5.3 相关工具.....87
9 安装与维护（适用于 1.0.0-1.0.10 版本）.....93
9.1 安装流程.....94
9.2 确认操作系统.....95
9.3 获取软件包.....96
9.3.1 检验软件包完整性.....96
9.4 创建运行用户.....96
9.5 安装驱动.....97
9.5.1 确认安装方式.....97
9.5.2 二进制文件直接安装.....99
9.5.2.1 安装驱动（适用于.run 格式）.....99
9.5.2.2 安装驱动（适用于.rpm 格式）.....104
9.5.3 源码编译安装.....107
9.5.3.1 安装依赖.....107
9.5.3.2 安装驱动（适用于.run 格式）.....110
9.5.4 重构驱动包安装.....115
9.5.4.1 安装依赖.....115
9.5.4.2 重构驱动包.....115
9.5.4.3 安装驱动（适用于.run 格式）.....116
9.6 安装固件.....119
9.6.1 安装固件（适用于.run 格式）.....120
9.6.2 安装固件（适用于.rpm 格式）.....121
9.7 卸载驱动.....123
9.7.1 卸载驱动（适用于.run 格式）.....123
9.7.2 卸载驱动（适用于.rpm 格式）.....124
9.8 卸载固件.....126
9.8.1 卸载固件（适用于.run 格式）.....126
9.8.2 卸载固件（适用于.rpm 格式）.....126
9.9 常用操作.....127
9.9.1 使用 WinSCP 传输文件.....127
9.9.2 设置用户有效期.....129
9.9.3 配置 Host 侧通过 SSH 方式登录 Device 的 IP.....130

9.9.4 修改用户密码 133
9.9.5 通过 PuTTY 登录（网口方式） 133
9.9.6 （可选）源码编译 135
9.9.6.1 源码包编译 135
9.9.6.1.1 获取和解压安装包 135
9.9.6.1.2 源码编译 136
9.9.6.1.3 源码编译安装 138
9.10 常见问题 138
9.10.1 Firmware 固件升级失败 139
9.10.2 CentOS 虚拟网口 IP 丢失 139
9.10.3 安装驱动过程中出现“drv_dkms_env_check failed, details in : /var/log/HiAI-install.log”的报错 140
9.10.4 升级系统内核版本时未卸载 run 包导致 run 包不可用 140
9.10.5 驱动重编译正常但安装上报 modprobe 失败 141
9.10.6 Centos7.6 系统宿主机目录挂载容器创建“/var/log/npu/slog/slogd”时，提示：Permission denied 142
9.10.7 Centos7.6 系统宿主机目录挂载容器拉起 slogd 时，提示：[WARNING] Init config path failed and continue 143
9.10.8 Ascend_monitor 通讯异常 143
9.10.9 安装驱动和固件后重启，“/var/log/npu/slog/host-0”日志中产生“[ERROR] raise script failed, cmd run failed, errno = 0”报错 144
9.10.10 安装驱动过程中出现“cp: cannot overwrite directory '/usr/local/sbin/npu-smi' with non-directory”的报错 145
9.10.11 安装驱动或启动服务过程中出现“please use cmd (lsattr /etc/sudoers) to check permission!”报错 146
9.10.12 Ascend 310 芯片无法正常启动，系统日志出现如下打印信息：dev id 0, The address dont comply with the NVME protocol(must be aligned with 4KB) 147
9.10.13 复位或下电导致 Device 无法正常启动 148
9.10.14 root 用户切换到普通用户需要输入密码权限 148
9.10.15 固件版本不一致导致服务器无法识别 Atlas 300I 推理卡 149
9.10.16 执行 lspci 命令出现“bash: lspci: command not found”报错 149
9.10.17 查询固件版本号后 Device 侧日志产生“check nve continer fail，ret : 258”报错 150
9.10.18 Centos 8.2 系统最小化安装导致驱动安装失败 150
9.11 参考 151
9.11.1 参数说明 151
10 工具 155
10.1 npu-smi 工具 155
10.1.1 安装 npu-smi 工具 155
10.1.2 npu-smi 命令介绍（适用于 5.1.RC 及以上版本） 156
10.1.3 npu-smi 命令介绍（适用于 1.0.11-1.0.15 版本） 156
10.1.3.1 简介 156
10.1.3.2 查询帮助 157
10.1.3.3 查询 npu-smi 工具版本 158
10.1.3.4 信息查询（info） 159
10.1.3.4.1 查询基本信息 159

10.1.3.4.2 查询详细信息 161
10.1.3.4.3 查询指定芯片信息 163
10.1.3.4.4 查询所有芯片监控数据 164
10.1.3.4.5 查询指定芯片监控数据 166
10.1.3.4.6 查询所有 NPU 设备 168
10.1.3.4.7 查询所有芯片映射关系信息 169
10.1.3.4.8 查询所有芯片常用信息 170
10.1.3.4.9 查询所有芯片闪存信息 172
10.1.3.4.10 查询指定芯片闪存信息 173
10.1.3.4.11 查询所有芯片内存信息 175
10.1.3.4.12 查询指定芯片内存信息 176
10.1.3.4.13 查询所有芯片统计信息 177
10.1.3.4.14 查询指定芯片统计信息 179
10.1.3.4.15 查询所有芯片温度 181
10.1.3.4.16 查询指定芯片温度 182
10.1.3.4.17 查询所有芯片功率 183
10.1.3.4.18 查询指定芯片功率 185
10.1.3.4.19 查询所有芯片电压 186
10.1.3.4.20 查询指定芯片电压 187
10.1.3.4.21 查询所有芯片健康状态 189
10.1.3.4.22 查询指定芯片健康状态 190
10.1.3.4.23 查询 MCU 监控信息 191
10.1.3.4.24 查询所有芯片的 ssh 使能状态 192
10.1.3.4.25 查询指定芯片的 ssh 使能状态 194
10.1.3.4.26 查询所有芯片 ECC 错误计数 195
10.1.3.4.27 查询指定芯片 ECC 错误计数 196
10.1.3.4.28 查询 ECC 使能状态 197
10.1.3.4.29 查询 p2p 使能状态 199
10.1.3.4.30 查询指定芯片 boot 区 200
10.1.3.4.31 查询指定芯片传感器信息 201
10.1.3.4.32 查询指定芯片系统时间 202
10.1.3.4.33 查询 NPU 与 MCU 之间的 IIC 通道是否正常 203
10.1.3.4.34 查询产品类型 204
10.1.3.4.35 查询指定芯片的产品类型 205
10.1.3.4.36 查询指定芯片的 Matrix 进程监控数据 206
10.1.3.4.37 查询用户证书 207
10.1.3.4.38 查询用户自定义信息 208
10.1.3.4.39 查询 AI 芯片的 IP 信息 209
10.1.3.4.40 查询所有芯片的 PCIE 误码计数 211
10.1.3.4.41 查询指定芯片的 PCIE 误码计数 213
10.1.3.4.42 查询所有芯片的容器共享模式 214
10.1.3.4.43 查询指定芯片的容器共享模式 215

10.1.3.4.44 查询指定芯片的算力等级 216
10.1.3.4.45 查询指定芯片的 AI CPU 和 Control CPU 的数量 217
10.1.3.4.46 查询所有芯片的 AI CPU、Control CPU 和 data CPU 数量 218
10.1.3.4.47 查询指定芯片的 AI CPU、Control CPU 和 data CPU 数量 219
10.1.3.4.48 查询指定芯片的虚拟 NPU 220
10.1.3.4.49 查询 NPU 设备首次上电的时间 221
10.1.3.4.50 查询所有芯片的进程占用内存 222
10.1.3.4.51 查询指定芯片的进程占用内存 224
10.1.3.4.52 查询指定芯片异步拷贝的业务数据长度 225
10.1.3.5 配置功能（set） 226
10.1.3.5.1 收集日志 226
10.1.3.5.2 复位标卡 227
10.1.3.5.3 复位芯片 229
10.1.3.5.4 设置 MCU 监控状态 231
10.1.3.5.5 设置 boot 区 232
10.1.3.5.6 设置机械硬盘上下电 233
10.1.3.5.7 配置所有芯片的 ssh 使能状态 234
10.1.3.5.8 配置指定芯片的 ssh 使能状态 236
10.1.3.5.9 配置 ECC 使能状态 237
10.1.3.5.10 配置用户证书 239
10.1.3.5.11 配置用户自定义信息 240
10.1.3.5.12 设置 AI 芯片的 IP 信息 241
10.1.3.5.13 清除所有芯片的 PCIE 误码计数 242
10.1.3.5.14 清除指定芯片的 PCIE 误码计数 244
10.1.3.5.15 设置清除所有芯片 ECC 错误计数 246
10.1.3.5.16 设置清除指定芯片 ECC 错误计数 247
10.1.3.5.17 设置所有芯片的容器共享模式 248
10.1.3.5.18 设置指定芯片的容器共享模式 249
10.1.3.5.19 设置指定芯片的算力等级 250
10.1.3.5.20 设置指定芯片的 AI CPU 数量 251
10.1.3.5.21 设置所有芯片的 AI CPU、Control CPU 和 data CPU 数量 252
10.1.3.5.22 设置指定芯片的 AI CPU、Control CPU 和 data CPU 数量 253
10.1.3.5.23 创建指定芯片的虚拟 NPU 254
10.1.3.5.24 销毁指定芯片的虚拟 NPU 256
10.1.3.5.25 秘钥吊销 257
10.1.3.5.26 设置指定芯片异步拷贝的业务数据长度 259
10.1.3.6 升级功能（upgrade） 260
10.1.3.6.1 查询固件升级状态 260
10.1.3.6.2 查询固件版本信息 261
10.1.3.6.3 升级固件 262
10.1.3.6.4 生效固件 264
10.1.3.7 清除功能（clear） 265

10.1.3.7.1 清除所有芯片 ECC 错误计数 265
10.1.3.7.2 清除指定芯片 ECC 错误计数 266
10.1.4 npu-smi 命令介绍（适用于 1.0.0-1.0.10 版本） 267
10.1.4.1 简介 267
10.1.4.2 npu-smi 使用方法 267
10.1.4.3 查询帮助 268
10.1.4.4 查询 npu-smi 工具版本 268
10.1.4.5 信息查询（info） 269
10.1.4.5.1 查询基本信息 269
10.1.4.5.2 查询芯片监控数据 270
10.1.4.5.3 查询所有 NPU 设备 271
10.1.4.5.4 查询所有芯片映射关系信息 271
10.1.4.5.5 查询具体卡信息 272
10.1.4.5.6 查询所有芯片常用信息 273
10.1.4.5.7 查询所有芯片闪存信息 274
10.1.4.5.8 查询所有芯片内存信息 275
10.1.4.5.9 查询所有芯片统计信息 276
10.1.4.5.10 查询所有芯片温度 277
10.1.4.5.11 查询所有芯片功率 278
10.1.4.5.12 查询所有芯片电压 278
10.1.4.5.13 查询所有芯片健康状态 279
10.1.4.5.14 查询所有芯片的 ssh 使能状态 280
10.1.4.5.15 查询所有芯片 ECC 错误计数 281
10.1.4.5.16 查询 ECC 使能状态 282
10.1.4.5.17 查询 p2p 使能状态 283
10.1.4.5.18 查询指定芯片信息 283
10.1.4.5.19 查询指定芯片温度 284
10.1.4.5.20 查询指定芯片功率 285
10.1.4.5.21 查询指定芯片电压 286
10.1.4.5.22 查询指定芯片统计信息 286
10.1.4.5.23 查询指定芯片闪存信息 287
10.1.4.5.24 查询指定芯片内存信息 287
10.1.4.5.25 查询指定芯片健康状态 288
10.1.4.5.26 查询指定芯片的 ssh 使能状态 289
10.1.4.5.27 查询指定芯片 ECC 错误计数 290
10.1.4.5.28 查询 NPU 与 MCU 之间的 IIC 通道是否正常 290
10.1.4.5.29 查询指定芯片的产品类型 291
10.1.4.5.30 查询指定芯片的 Matrix 进程监控数据 292
10.1.4.5.31 查询用户证书 293
10.1.4.5.32 查询 AI 芯片的 IP 信息 293
10.1.4.5.33 查询所有芯片的 PCIE 误码计数 294
10.1.4.5.34 查询指定芯片的 PCIE 误码计数 295

10.1.4.6 配置功能（set） 295
10.1.4.6.1 配置所有芯片的 ssh 使能状态 296
10.1.4.6.2 配置指定芯片的 ssh 使能状态 297
10.1.4.6.3 配置 ECC 使能状态 298
10.1.4.6.4 配置 p2p 使能状态 298
10.1.4.6.5 收集日志 299
10.1.4.6.6 复位标卡 300
10.1.4.6.7 复位芯片 302
10.1.4.6.8 配置用户证书 303
10.1.4.6.9 设置 AI 芯片的 IP 信息 304
10.1.4.6.10 清除所有芯片的 PCIE 误码计数 305
10.1.4.6.11 清除指定芯片的 PCIE 误码计数 306
10.1.4.7 升级功能（upgrade） 307
10.1.4.7.1 查询固件升级状态 307
10.1.4.7.2 查询固件版本信息 308
10.1.4.7.3 升级固件 308
10.1.4.7.4 生效固件 309
10.1.5 卸载 npu-smi 工具 310
10.2 NPU 一键式日志收集工具 311
10.3 驱动一键式收集日志工具 312
11 维护管理 315
11.1 带内管理 315
11.2 带外管理 315
12 容器内安装 316
12.1 简介 316
12.2 宿主机目录挂载容器 317
12.3 宿主机目录不挂载容器 320
12.4 容器内软件包卸载 323
A 附录 324
A.1 术语 324
A.2 缩略语 325
A.3 kubelet 代码实现 326
A.4 免责声明 328
A.5 如何获取帮助 329
A.5.1 收集必要的故障信息 329
A.5.2 做好必要的调试准备 329
A.5.3 如何使用文档 329
A.5.4 获取技术支持 329

1 安全

1.1 通用安全注意事项
1.2 设备上的标志
1.3 电气安全

1.1 通用安全注意事项

在安装、操作、维护华为公司制造的设备时，本文档介绍的所应遵守的部分安全注意事项可指导选择测量设备和测试设备。

所有安全注意事项

为保障人身和设备安全，在安装、操作和维护设备时，请遵循设备上标识及手册中说明的所有安全注意事项。

手册中的“注意”、“警告”和“危险”事项，并不代表所应遵守的所有安全事项，只作为所有安全注意事项的补充。

当地法规和规范

操作设备时，应遵守当地法规和规范。手册中的安全注意事项仅作为当地安全规范的补充。

基本安装要求

负责安装维护华为设备的人员，必须先经严格培训，了解各种安全注意事项，掌握正确的操作方法之后，方可安装、操作和维护设备。

- 只允许有资格和培训过的人员安装、操作和维护设备。
- 只允许有资格的专业人员拆除安全设施和检修设备。
- 替换和变更设备或部件（包括软件）必须由华为认证或授权的人员完成。
- 操作人员应及时向负责人汇报可能导致安全问题的故障或错误。

接地要求

以下要求只针对需要接地的设备：

- 安装设备时，必须先接地；拆除设备时，最后再拆地线。
- 禁止破坏接地导体。
- 禁止在未安装接地导体时操作设备。
- 设备应永久性地接到保护地。操作设备前，应检查设备的电气连接，确保设备已可靠接地。

人身安全

- 操作设备前，应穿防静电工作服，佩戴防静电手套或腕带，并去除首饰和手表等易导电物体，以免被电击或灼伤。
- 如果发生火灾，应撤离建筑物或设备区域并按下火警警铃，或者拨打火警电话。任何情况下，严禁再次进入燃烧的建筑物。

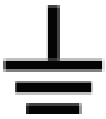
设备安全

- 操作前，应先将设备可靠地固定在地板或其他稳固的物体上，如墙体或安装架。
- 系统运行时，请勿堵塞通风口。
- 安装面板时，如果螺钉需要拧紧，必须使用工具操作。
- 安装完设备，请清除设备区域的空包装材料。

1.2 设备上的标志

介绍设备上的标志，有警告标志、接地标志和防静电标志。

表 1-1 安全标志

图示	名称	说明
	警告标志	该标志表示误操作可能会导致设备损坏或人身伤害。
	外部接地标志	该标志是设备外部的接地标识。接地电缆的两端分别接在不同设备上，表示设备必须通过接地点接地，保证设备能够正常运行，同时保证操作人员的人身安全。
	内部接地标志	该标志是设备内部的接地标识。接地电缆的两端都接在同一个设备上的不同组件上，表示设备必须通过接地点接地，保证设备能够正常运行，同时保证操作人员的人身安全。
	防静电标志	该标志表示为静电敏感区，请勿徒手触摸设备。在该区域操作时，请采取严格的防静电措施，例如佩戴防静电腕带或者防静电手套。

1.3 电气安全

介绍电源线、静电放电的安全注意事项。

高压

危险

- 高压电源为设备的运行提供电力，直接接触或通过潮湿物体间接接触高压电源，会带来致命危险。
- 不规范、不正确的高压操作，会引起火灾或电击等意外事故。

雷雨天气

此要求仅适用于无线基站或带有天馈线的设备。

危险

禁止在雷雨天气下进行高压、交流电操作及铁塔、桅杆作业，否则会有生命危险。

大漏电流

危险

- 在接通电源之前设备必须先接地，否则会危及人身及设备安全。
- 如果设备电源端子附近粘贴有“大漏电流”标志，在连接交流输入电源之前，必须先将设备机壳的保护接地端子接地，以防止设备的漏电流对人体产生电击。

电源线

危险

- 禁止带电安装或拆除电源线。电源线芯在接触导体的瞬间，会产生电弧或电火花，可导致火灾或眼睛受伤。
- 安装、拆除电源线之前，必须先关闭电源开关。
- 连接电源线之前，必须先确认电源线标签标识正确再进行连接。

保险丝

须知

为保证设备运行安全，当设备上的保险丝熔断后，应使用相同型号和规格的保险丝替换。

静电放电

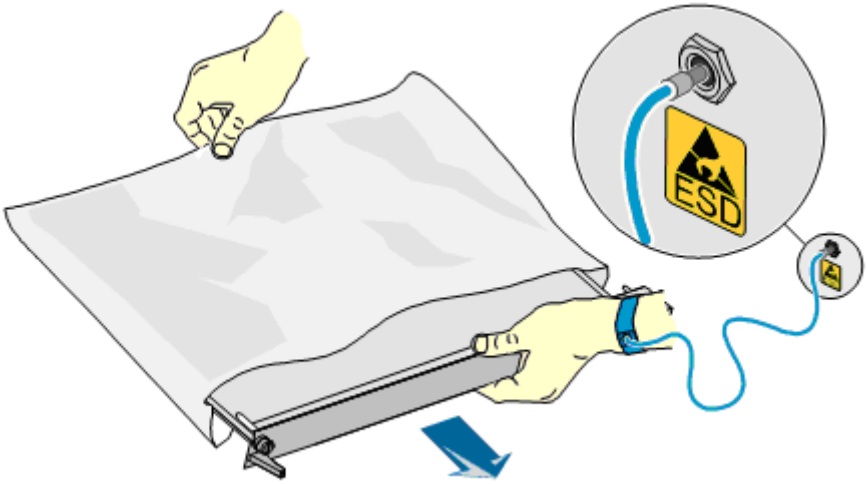
须知

人体产生的静电会损坏单板上的静电敏感元器件，如大规模集成电路（LSI）等。

- 在人体移动、衣服摩擦、鞋与地板的摩擦或手拿普通塑料制品等情况下，人体会产生静电电磁场，在放电前不易消失。
- 在接触设备，手拿单板或专用集成电路（ASIC）芯片等之前，为防止人体静电损坏敏感元器件，必须佩戴防静电腕带，并将防静电腕带的另一端良好接地。

  防静电腕带佩戴如图1-1所示。

  图 1-1 佩戴防静电腕带示意图



2 产品简介

2.1 概述
2.2 外观
2.3 系统框图

2.1 概述

Atlas 300I 推理卡（型号 3000）采用4个昇腾310 AI处理器的PCIE HHHL卡，配合主设备实现快速高效的推理计算、图像识别及视频处理等工作。

说明

- 昇腾310 AI处理器是一款专门为推理计算、图像识别、视频处理及机器学习等领域设计的高性能、低功耗AI芯片。芯片内置2个AI core，可支持128位宽的LPDDR4X，最高可提供22TOPS INT8/PCS的计算能力。
- 推理卡只支持用于AI推理任务，不支持训练任务。

2.2 外观

Atlas 300I 推理卡（型号 3000）外观如图2-1和图2-2所示。

图 2-1 Atlas 300I 推理卡（型号 3000）半高拉手条外观图

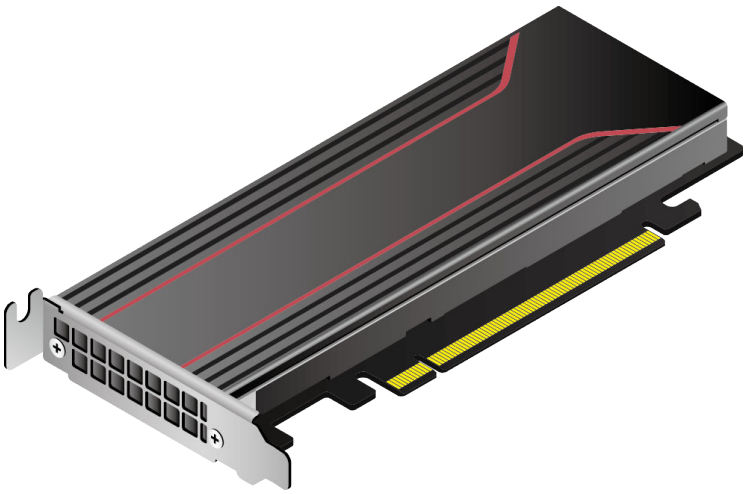
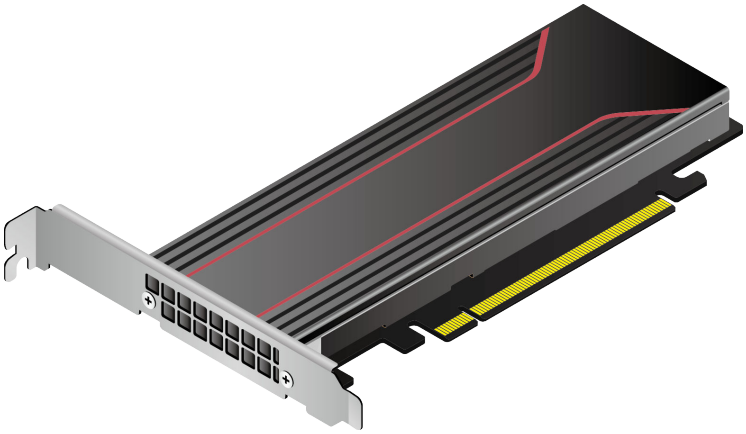


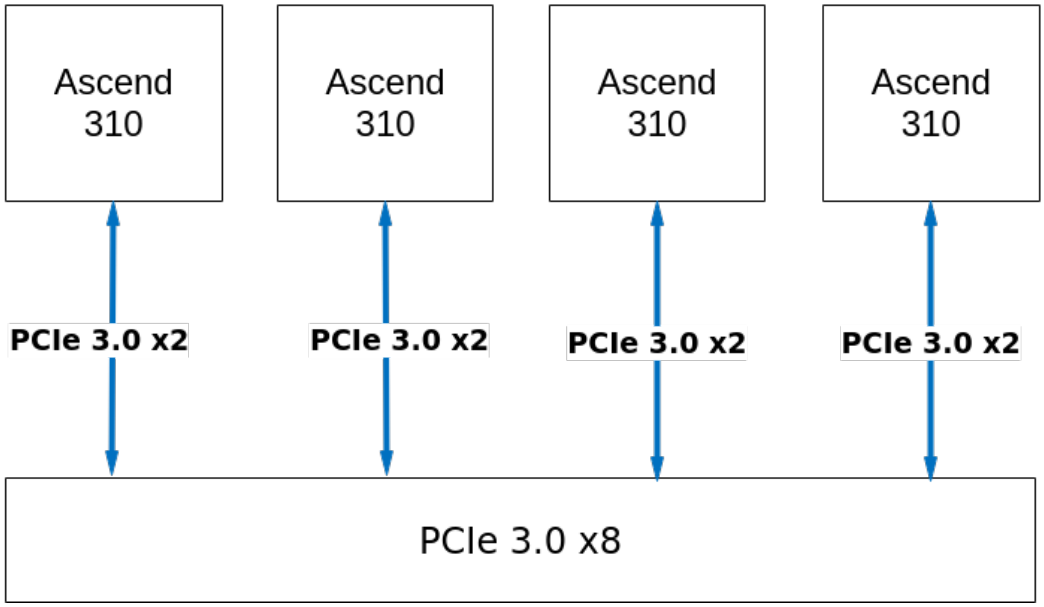
图 2-2 Atlas 300I 推理卡（型号 3000）全高拉手条外观图



2.3 系统框图

Atlas 300I 推理卡（型号 3000）的系统框图如图2-3所示。

图 2-3 Atlas 300I 推理卡（型号 3000）系统框图



3 产品特点

3.1 性能特点
3.2 可维护性特点
3.3 典型应用场景

3.1 性能特点

- 采用4个高性能低功耗的昇腾310 AI处理器，最高可提供22TOPS INT8/PCS的计算能力。
- 支持多种规格的H.264、H.265视频编解码，适用于用户不同的视频处理需求。

3.2 可维护性特点

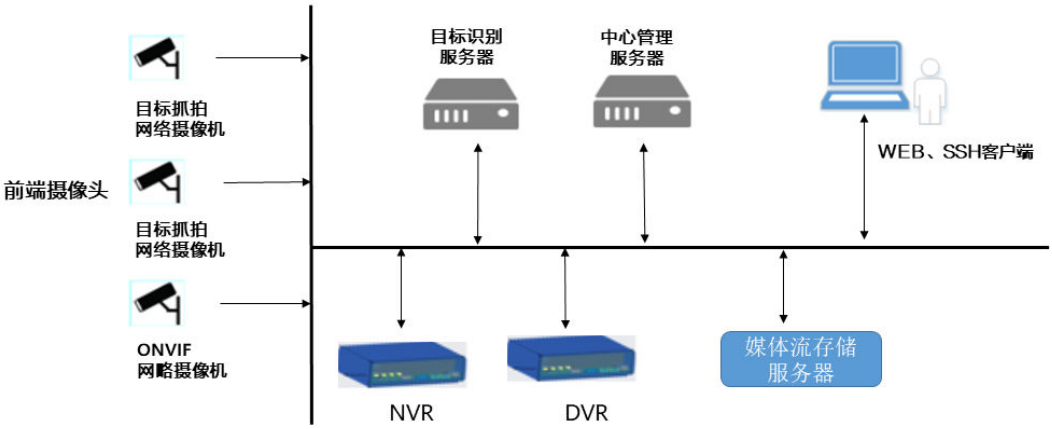
- 支持带内的在线升级，方便客户进行日常维护。
- 支持带内及带外获取温度、电压、功耗等设备状态信息，提供的图形界面易于管理。
- 完备的命令行管理功能，用户可以通过各种命令进行日常的设备管理。
- 支持带内及带外资产管理功能，提供序列号等信息，方便资产管理。

3.3 典型应用场景

Atlas 300I 推理卡（型号 3000）典型应用场景为目标识别系统，系统主要采用了目标检测算法、目标跟踪算法、目标质量评分算法以及高速目标对比识别算法，实现了实时目标抓拍建模、实时黑名单对比报警和目标后台检索等功能。

目标识别系统架构如**图3-1**所示，主要部件有系统前端高清网络摄像机或目标抓拍机、媒体流存储服务器（可选）、目标智能分析服务器、目标比对搜索服务器、中心管理服务器、客户端管理软件等组成。Atlas 300I 推理卡（型号 3000）部署在目标智能分析服务器中，主要实现视频解码/预处理、目标检测推理，目标对齐（矫正）和目标特征提取推理功能。

图 3-1 典型的目标识别系统架构图



4 产品规格

4.1 基本规格
4.2 环境条件
4.3 时钟要求
4.4 电源管理
4.5 热插拔
4.6 散热规格

4.1 基本规格

表 4-1 Atlas 300I 推理卡（型号 3000）规格

特征	规格
形态	半高半长PCIe标卡，支持全高和半高两种拉手条
AI处理器	4 x 昇腾310 AI处理器，每个处理器包含： ● 2个DaVinci AI Core ● 8个A55 Arm Core（最大主频1.6GHz）
AI算力[a]	● 半精度（FP16）：44 TFLOPS ● 整数精度（INT8）：88 TOPS
内存	● LPDDR4X ● 容量：32GB ● 位宽：512bit ● 速率：3200Mbps ● 总带宽：204.8GByte/s ● 支持ECC **说明** 每个处理器容量为8G，位宽为128bit，总带宽为51.2GByte/s。

特征	规格
编解码能力	● 支持H.264/H.265 Decoder硬件解码，80路1080P (1920 x 1080) 25FPS，YUV420 ● 支持H.264/H.265 Decoder硬件解码，64路1080P (1920 x 1080) 30FPS，YUV420 ● 支持H.264/H.265 Decoder硬件解码，8路4K (3840 x 2160) 60FPS，YUV420 ● 支持H.264/H.265 Encoder硬件编码，4路1080P (1920 x 1080) 30FPS，YUV420 ● JPEG解码能力4 x1080P (1920 x 1080) 256FPS，编码能力4 x1080P (1920 x 1080) 64FPS，最大分辨率：8192 x 4320 ● PNG解码能力4 x1080P (1920 x 1080) 24FPS，最大分辨率：4096 x 2160
PCIe接口	PCIe 3.0 x8，兼容PCIe 2.0/PCIe 1.0 **说明** ● PCIe接口不是标准x8接口，硬件形态是x16金手指，信号定义是由4个x2端口合并。 ● 服务器端需要做端口分配，将标准x8端口分为4个独立的x2端口。
PCI IDs	Vendor ID: 0x19E5 Device ID: 0xD100 Subsystem Vendor ID: 0x0200 Subsystem Device ID: 0x0100
功耗	最大功耗为67W
尺寸（长×高×宽）	169.5mm × 68.9mm × 18.5mm
净重	310g
a：稳定提供的峰值算力。	

4.2 环境条件

Atlas 300I 推理卡（型号 3000）硬件应用环境条件如表4-2所示。

表 4-2 Atlas 300I 推理卡（型号 3000）硬件应用环境条件

环境指标	规格
温度	● 工作温度：0℃～55℃（32℉～131℉） ● 存储温度：-40℃～+75℃（-40℉～+167℉）
湿度	● 工作湿度：5%RH～90%RH（非冷凝） ● 存储湿度：5%RH～95%RH（非冷凝）

环境指标	规格
海拔高度	小于3000m。高于900m使用时，海拔每升高300m最高温度规格降低1℃。 **说明** 按照ASHRAE 2015标准： ● 配置满足ASHRAE Class A1、A2时，海拔高度超过900m，工作温度按每升高300m降低1°C计算。 ● 配置满足ASHRAE Class A3时，海拔高度超过900m，工作温度按每升高175m降低1°C计算。 ● 配置满足ASHRAE Class A4时，海拔高度超过900m，工作温度按每升高125m降低1°C计算。

4.3 时钟要求

Atlas 300I 推理卡（型号 3000）遵从标准PCIe标卡协议（PCI Express® Card Electromechanical Specification Revision 3.0），整卡只需要提供一路标准PCIe 3.0（可向下兼容 2.0及1.0）差分时钟，信号质量满足PCIe规范。

4.4 电源管理

Atlas 300I 推理卡（型号 3000）遵从标准PCIe标卡协议（PCI Express® Card Electromechanical Specification Revision 3.0），整卡最大功耗为67W，要求对应Atlas 300I 推理卡（型号 3000）槽位可提供5.5A@12V及0.5A@3.3V标准供电能力。

4.5 热插拔

Atlas 300I 推理卡（型号 3000）不支持通知式热插拔和暴力热插拔。

4.6 散热规格

4.6.1 散热要求

Atlas 300I 推理卡（型号 3000）用于带风扇的主动散热环境，支持双向进风出风，风量必须满足散热要求。

表 4-3 Atlas 300I 推理卡（型号 3000）散热要求

卡入风口平均温度/℃	卡需求最低风量/CFM	压降/Inch H_2O
55	6.7	0.53
50	5.3	0.37
45	4.4	0.28
40	3.7	0.21

卡入风口平均温度/℃	卡需求最低风量/CFM	压降/Inch H_2O
35	3.3	0.18
30	3.0	0.16
最低散热要求	3.0	0.16

说明

- 卡需求的最低风量为通过散热器风量。
- 散热器入口环境温度为进风口的平均温度。
- 需求的风量是建议值，不同系统提供给Atlas 300I 推理卡（型号 3000）的风量和温度可能存在差异，需要进行实测确定。
- Atlas 300I 推理卡（型号 3000）上电状态，系统需要提供基础散热能力，需求的最低风量为3.0CFM。

4.6.2 散热规格

Atlas 300I 推理卡（型号 3000）支持入口温度为0℃～55℃，内部有温度监控点，带内及带外均可对Ascend 310、存储芯片进行实时监控。系统的散热条件需保证该卡在工作过程中实际温度值低于规格值。

表 4-4 关键器件温度规格

规格	Ascend 310温度 ℃	存储芯片温度 ℃
下电温度阈值	106	100
降频温度阈值	101	90
长期工作温度阈值	100	85

4.6.3 过温保护

Atlas 300I 推理卡（型号 3000）支持带外及带内通道检测Ascend 310及存储芯片等关键器件的结温，同时也支持检测整板温度，Atlas 300I 推理卡（型号 3000）日志中会记录最高温度和过温的次数及总时间等信息。

Atlas 300I 推理卡（型号 3000）的主要器件Ascend 310及其存储芯片最高支持入风口温度为55℃，为保证可靠工作，外界需提供散热需求的风量，并设计了如下温控策略，Atlas 300I 推理卡（型号 3000）使用了2级预警机制：

- 第一级为严重告警，Ascend 310芯片的严重告警阈值为101℃，存储芯片的严重告警阈值为90℃。当芯片结温或环境温度达到该值，固件就会限制设备的性能。

- 第二级为致命告警，Ascend 310芯片的致命告警阈值为106℃，存储芯片的致命告警阈值为100℃。当芯片结温或环境温度达到该值，Atlas 300I 推理卡（型号3000）会启动自身下电。

5 硬件兼容性

- Atlas 300I 推理卡（型号 3000）支持的华为服务器型号：Atlas 500 Pro 智能边缘服务器（型号：3000）、Atlas 800 推理服务器（型号：3000）。
- Atlas 300I 推理卡（型号 3000）支持的第三方服务器型号，详细信息可登录**昇腾兼容性认证服务器目录**，使用产品型号进行查询。

6 信号管脚

6.1 管脚定义

6.1 管脚定义

Atlas 300I 推理卡（型号 3000）遵从标准PCIe标卡协议（PCI Express® Card Electromechanical Specification Revision 3.0），对外提供PCIe X16金手指物理接口，主要提供的数据信号为：一组PCIe 3.0 x8（分为四组PCIe 3.0 x2），最大速率8Gbps/lane，用于业务数据的交互传递，一组SMBUS，最大速率100Kbps，用于带外管理单元监控Atlas 300I 推理卡（型号 3000），Atlas 300I 推理卡（型号 3000）详细信号分配如**表6-1** ~ **表6-5**所示。

表 6-1 Atlas 300I 推理卡（型号 3000）金手指管脚定义（Mechanical key）

序号		管脚名	描述	备注
Side B	1	+12V	12V电源	12V电源
	2	+12V	12V电源	
	3	+12V	12V电源	
	4	GND	地	接地
	5	SMCLK	SMBus时钟	输入
	6	SMDAT	SMBus数据	输入
	7	GND	地	接地
	8	+3.3V	3.3V电源	3.3V电源
	9	JTAG1	JTAG接口TRST信号	悬空
	10	3.3Vaux	3.3V auxiliary电源	
	11	WAKE#	链接重新激活的信号	

序号		管脚名	描述	备注
Side A	1	PRSNT1#	热插拔存在检测1#	在位检测
	2	+12V	12V电源	12V电源
	3	+12V	12V电源	
	4	GND	地	接地
	5	JTAG2	JTAG接口TCK信号	悬空
	6	JTAG3	JTAG接口TDI信号	
	7	JTAG4	JTAG接口TDO信号	
	8	JTAG5	JTAG接口TMS信号	
	9	+3.3V	3.3V电源	3.3V电源
	10	+3.3V	3.3V电源	
	11	PERST#	基本复位	整板复位

表 6-2 Atlas 300I 推理卡（型号 3000）金手指管脚定义（End of the x1 connector）

序号		管脚名	描述	备注
Side B	12	RSVD	预留	悬空
	13	GND	地	接地
	14	PETp0	发送差分对PCIe_TX_0	接Ascend 310 1：TX0
	15	PETn0		
	16	GND	地	接地
	17	PRSNT2#	热插拔存在检测2#	悬空
	18	GND	地	接地
Side A	12	GND	地	接地
	13	REFCLK+	差分时钟	底板提供100M PCIe差分时钟，支持3.0/2.0/1.0，支持SSC
	14	REFCLK-		
	15	GND	地	接地
	16	PERp0	接收差分对PCIe_RX_0	接Ascend 310 1：RX0
	17	PERn0		
	18	GND	地	接地

表 6-3 Atlas 300I 推理卡（型号 3000）金手指管脚定义（End of the x4 connector）

序号		管脚名	描述	备注
Side B	19	PETp1	发送差分对 PCIe_TX_1	接Ascend 310 1：TX1
	20	PETn1		
	21	GND	地	接地
	22	GND	地	接地
	23	PETp2	发送差分对 PCIe_TX_2	接Ascend 310 2：TX0
	24	PETn2		
	25	GND	地	接地
	26	GND	地	接地
	27	PETp3	发送差分对 PCIe_TX_3	接Ascend 310 2：TX1
	28	PETn3		
	29	GND	地	接地
	30	RSVD	预留	悬空
	31	PRSNT2#	热插拔存在检测2#	悬空
	32	GND	地	接地
Side A	19	RSVD	悬空	
	20	GND	地	接地
	21	PERp1	接收差分对 PCIe_RX_1	接Ascend 310 1：RX1
	22	PERn1		
	23	GND	地	接地
	24	GND	地	接地
	25	PERp2	接收差分对 PCIe_RX_2	接Ascend 310 2：RX0
	26	PERn2		
	27	GND	地	接地
	28	GND	地	接地
	29	PERp3	接收差分对 PCIe_RX_3	接Ascend 310 2：RX1
	30	PERn3		
	31	GND	地	接地
	32	RSVD	预留	悬空

表 6-4 Atlas 300I 推理卡（型号 3000）金手指管脚定义（End of the x8 connector）

序号		管脚名	描述	备注
Side B	33	PETp4	发送差分对PCIe_TX_4	接Ascend 310 3：TX0
	34	PETn4		
	35	GND	地	接地
	36	GND	地	接地
	37	PETp5	发送差分对PCIe_TX_5	接Ascend 310 3：TX1
	38	PETn5		
	39	GND	地	接地
	40	GND	地	接地
	41	PETp6	发送差分对PCIe_TX_6	接Ascend 310 4：TX0
	42	PETn6		
	43	GND	地	接地
	44	GND	地	接地
	45	PETp7	发送差分对PCIe_TX_7	接Ascend 310 4：TX1
	46	PETn7		
	47	GND	地	接地
	48	PRSNT2#	热插拔存在检测2#	悬空
	49	GND	地	接地
Side A	33	RSVD	预留	悬空
	34	GND	地	接地
	35	PERp4	接收差分对PCIe_RX_4	接Ascend 310 3：RX0
	36	PERn4		
	37	GND	地	接地
	38	GND	地	接地
	39	PERp5	接收差分对PCIe_RX_5	接Ascend 310 3：RX1
	40	PERn5		
	41	GND	地	接地
	42	GND	地	接地
	43	PERp6	接收差分对PCIe_RX_6	接Ascend 310 4：RX0
	44	PERn6		

序号		管脚名	描述	备注
	45	GND	地	接地
	46	GND	地	接地
	47	PERp7	接收差分对PCIe_RX_7	接Ascend 310 4：RX1
	48	PERn7		
	49	GND	地	接地

表 6-5 Atlas 300I 推理卡（型号 3000）金手指管脚定义（End of the x16 connector）

序号		管脚名	描述	备注
Side B	50	PETp8	发送差分对PCIe_TX_8	悬空
	51	PETn8		
	52	GND	地	接地
	53	GND	地	接地
	54	PETp9	发送差分对PCIe_TX_9	悬空
	55	PETn9		
	56	GND	地	接地
	57	GND	地	接地
	58	PETp10	发送差分对PCIe_TX_10	悬空
	59	PETn10		
	60	GND	地	接地
	61	GND	地	接地
	62	PETp11	发送差分对PCIe_TX_11	悬空
	63	PETn11		
	64	GND	地	接地
	65	GND	地	接地
	66	PETp12	发送差分对PCIe_TX_12	悬空
	67	PETn12		
	68	GND	地	接地
	69	GND	地	接地

序号		管脚名	描述	备注
	70	PETp13	发送差分对PCIe_TX_13	悬空
	71	PETn13		
	72	GND	地	接地
	73	GND	地	接地
	74	PETp14	发送差分对PCIe_TX_14	悬空
	75	PETn14		
	76	GND	地	接地
	77	GND	地	接地
	78	PETp15	发送差分对PCIe_TX_15	悬空
	79	PETn15		
	80	GND	地	接地
	81	PRSNT2#	热插拔存在检测2#	PCIe卡上连接
	82	RSVD	预留	悬空
Side A	50	RSVD	预留	悬空
	51	GND	地	接地
	52	PERp8	接收差分对PCIe_RX_8	悬空
	53	PERn8		
	54	GND	地	接地
	55	GND	地	接地
	56	PERp9	接收差分对PCIe_RX_9	悬空
	57	PERn9		
	58	GND	地	接地
	59	GND	地	接地
	60	PERp10	接收差分对PCIe_RX_10	悬空
	61	PERn10		
	62	GND	地	接地
	63	GND	地	接地
	64	PERp11	接收差分对PCIe_RX_11	悬空
	65	PERn11		
	66	GND	地	接地

序号		管脚名	描述	备注
	67	GND	地	接地
	68	PERp12	接收差分对 PCIe_RX_12	悬空
	69	PERn12		
	70	GND	地	接地
	71	GND	地	接地
	72	PERp13	接收差分对 PCIe_RX_13	悬空
	73	PERn13		
	74	GND	地	接地
	75	GND	地	接地
	76	PERp14	接收差分对 PCIe_RX_14	悬空
	77	PERn14		
	78	GND	地	接地
	79	GND	地	接地
	80	PERp15	接收差分对 PCIe_RX_15	悬空
	81	PERn15		
	82	GND	地	接地

版权所有 © 华为技术有限公司

7 安装与维护（适用于 5.1.RC2 及以上版本）

驱动和固件的安装与维护请参见**《Atlas 300I 推理卡 NPU驱动和固件安装指南（型号 3000）》**。

版权所有 © 华为技术有限公司

8 安装与维护（适用于 1.0.11-1.0.15 版本）

8.1 用户必读
8.2 物理机安装
8.3 常用操作
8.4 FAQ
8.5 参考

8.1 用户必读

软件包命名

- 1.0.11版本软件包命名如下：
  - 本文中的驱动包命名：*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run、*{product name}*-npu-driver-*x.x.x*-linux.*{arch}*.rpm、*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.deb
  - 本文中的固件包命名：*{product name}*-npu-firmware_*<version>*.run、*{product name}*-npu-firmware-*<version>*.rpm、*{product name}*-npu-firmware_*<version>*.deb
- 1.0.12及以上版本软件包命名如下：
  - 本文中的驱动包命名：*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run、*{product name}*-npu-driver-*x.x.x*-*<release>*.*{arch}*.rpm、*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.deb
  - 本文中的固件包命名：*{product name}*-npu-firmware_*<version>*.run、*{product name}*-npu-firmware-*<version>*-*<release>*.noarch.rpm、*{product name}*-npu-firmware_*<version>*.deb
- 其中 *{product name}*表示产品名称，*x.x.x*表示NPU驱动版本号，*<release>*表示RPM包的发布版本号，*<version>*表示芯片固件版本号，*{arch}*表示系统架构，具体请根据实际情况进行替换。

  {product name} 具体包含以下名称：
  - A300-3000
  - A300-3000-3010

版权所有 © 华为技术有限公司

注意事项

不支持run包、rpm包和deb包混用安装或升级。若用户已使用其中一种软件包安装，想要切换为其他类型的软件包，则需要先使用对应类型软件包的卸载命令进行卸载，再重新安装同版本新软件包。

若用户误操作run包、rpm包和deb包的混用安装或升级，可以参考**8.4.6 软件包格式混用安装或升级恢复方法**。

安装场景说明

表 8-1 安装场景说明

安装说明	参考章节
如果用户推理服务部署在物理机，则在物理机安装驱动固件即可。	**8.2 物理机安装**
如果用户推理服务部署在容器，则需先在物理机安装驱动固件，再在容器内安装驱动。	**8.2 物理机安装** **12 容器内安装**

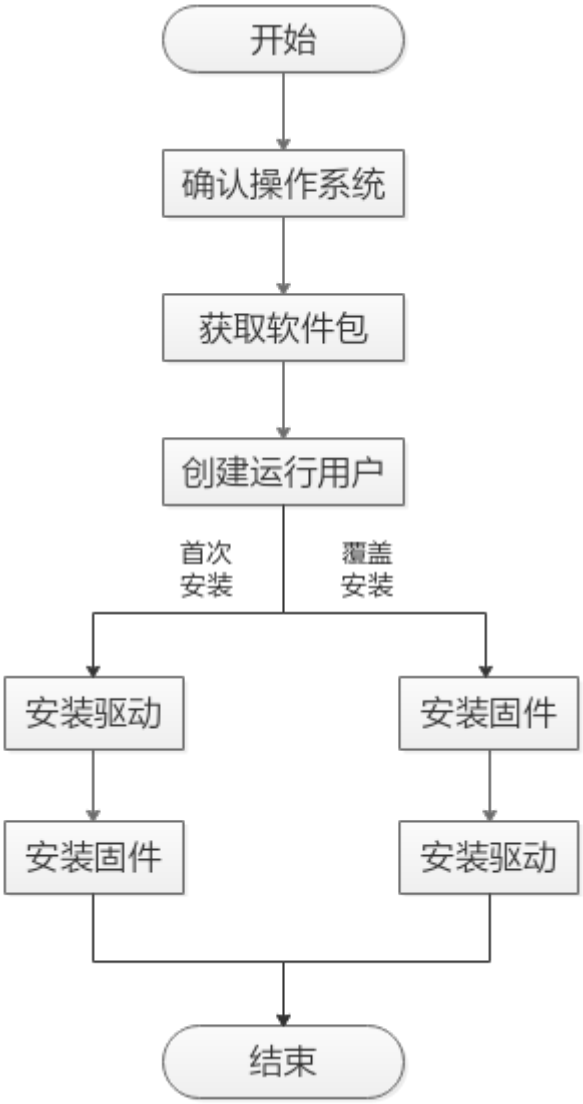
8.2 物理机安装

本章节主要介绍Linux系统下驱动和固件的安装与卸载。

8.2.1 安装流程

安装流程

图 8-1 软件包安装流程



说明

首次安装场景：硬件设备刚出厂时未安装驱动，或者硬件设备前期安装过驱动固件但是当前已卸载，上述场景属于首次安装场景，需按照“**驱动->固件**”的顺序安装驱动固件。

覆盖安装场景：硬件设备前期安装过驱动固件且未卸载，当前要再次安装驱动固件，此场景属于覆盖安装场景，需按照“**固件->驱动**”的顺序安装固件驱动。

表 8-2 安装流程说明

关键步骤	说明	参考章节
确认操作系统	确认目标环境支持的操作系统。	8.2.2 确认操作系统

关键步骤	说明	参考章节
获取软件包	根据目标环境所使用的操作系统下载所需软件包。	8.2.3 获取软件包
创建运行用户	使用root用户安装软件包，但安装完之后要求使用root用户（该场景下权限控制可能存在安全风险）或非root用户（不能为root用户属组）运行相关进程。	8.2.4 创建运行用户
安装驱动	支持以下三种安装场景。 ● 二进制文件直接安装 ● 源码编译安装 ● 重构驱动包安装 **说明** 首次安装请按照“**驱动->固件**”的顺序；覆盖安装或升级请按照“**固件->驱动**”的顺序，分别安装软件包。	8.2.5 安装驱动
安装固件	根据不同的软件包格式指导用户安装固件。 **说明** 首次安装请按照“**驱动->固件**”的顺序；覆盖安装或升级请按照“**固件->驱动**”的顺序，分别安装软件包。	8.2.6 安装固件

8.2.2 确认操作系统

操作步骤

步骤1 查询服务器及配套PCIe卡支持的操作系统，详细请参考计算产品兼容性查询助手。

- 若使用华为服务器，则“产品型号”选择对应服务器型号，“PCIe卡”信息选择对应卡型号，单击“查询”。
- 若使用第三方服务器，则“产品型号”选择对应卡型号，单击“查询”。

步骤2 执行如下命令，查询服务器当前运行环境的操作系统架构及版本。

```
uname -m && cat /etc/*release
```

说明

如果查询的操作系统版本不在兼容性列表中，请替换为兼容性列表中支持的操作系统。

----结束

8.2.3 获取软件包

操作步骤

- 企业网用户：

a. 登录**A300-3000**的软件页签。

b. 选择目标版本。

c. 查看《版本配套表》，根据“配套表”页签中的“配套操作系统版本”列说明选择所需的软件包。

说明

驱动和固件包格式支持run、rpm、deb，每种格式软件包安装效果无区别，但是支持的OS范围有区别，请用户根据现场OS系统需求选择软件包。

一种格式的软件包可能支持多种OS，如Ubuntu OS支持run和deb格式，用户可以任意选择一种格式的软件包，如果用户对软件包格式无特殊需求，那么在OS支持的前提下优先推荐下载run格式的软件包。

d. 单击软件包后面的和，可获取软件包和数字签名文件。

说明

请获取同一版本下的驱动包、芯片固件包、MCU固件包和驱动包重新打包工具。

- 运营商用户：联系华为技术有限公司驻当地办事处的技术支持人员。

软件数字签名验证

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：**http://support.huawei.com/carrier/digitalSignatureAction**

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

8.2.4 创建运行用户

首先介绍安装用户和运行用户概念，安装用户为安装驱动和固件所使用的用户，运行用户为驱动固件安装完成后，后续运行推理或训练业务时启动运行驱动和固件的用户。

驱动和固件的安装用户必须为root用户，驱动和固件的运行用户可以为root用户或者非root用户，如果后续用户对运行业务有用户权限控制或者多用户隔离的需求，可创建非root用户作为后续运行用户。本章节介绍创建运行用户的步骤。

若使用rpm包安装时，系统会自动创建用户组和用户HwHiAiUser，用户无需手动创建。请跳过此章节。

若使用FusionDirector工具纳管，首次安装驱动包（.zip格式）系统会自动创建用户组和用户HwHiAiUser，无需手动创建。请跳过此章节。

表 8-3 创建运行用户

运行用户	操作步骤

• root用户
• 非root用户（推荐）
  – HwHiAiUser
  – 非HwHiAiUser
说明
如果使用root用户作为运行用户，则需要与--install-for-all参数配合使用，详情请参见**8.5.1 参数说明/常用命令**，该场景下权限控制可能存在安全风险；如果使用非root用户作为运行用户，则创建的该用户不能为root用户属组。
	• 如果创建的用户是HwHiAiUser，安装软件包时无需指定运行用户，默认即为HwHiAiUser。
	• 如果创建的用户是非HwHiAiUser（含root），安装软件包时必须指定运行用户（通过--install-username=*username* --install-usergroup=*usergroup*参数指定）。因此如果对运行用户名称没有特殊要求，建议使用HwHiAiUser。
	请参见如下方法创建运行用户。
	1. 以root用户登录服务器。
	2. 执行如下命令，创建运行用户。
	# groupadd *usergroup*
	# useradd -g *usergroup* **-d /home/**_username_ **-m** *username* **-s /bin/bash**

8.2.5 安装驱动

须知

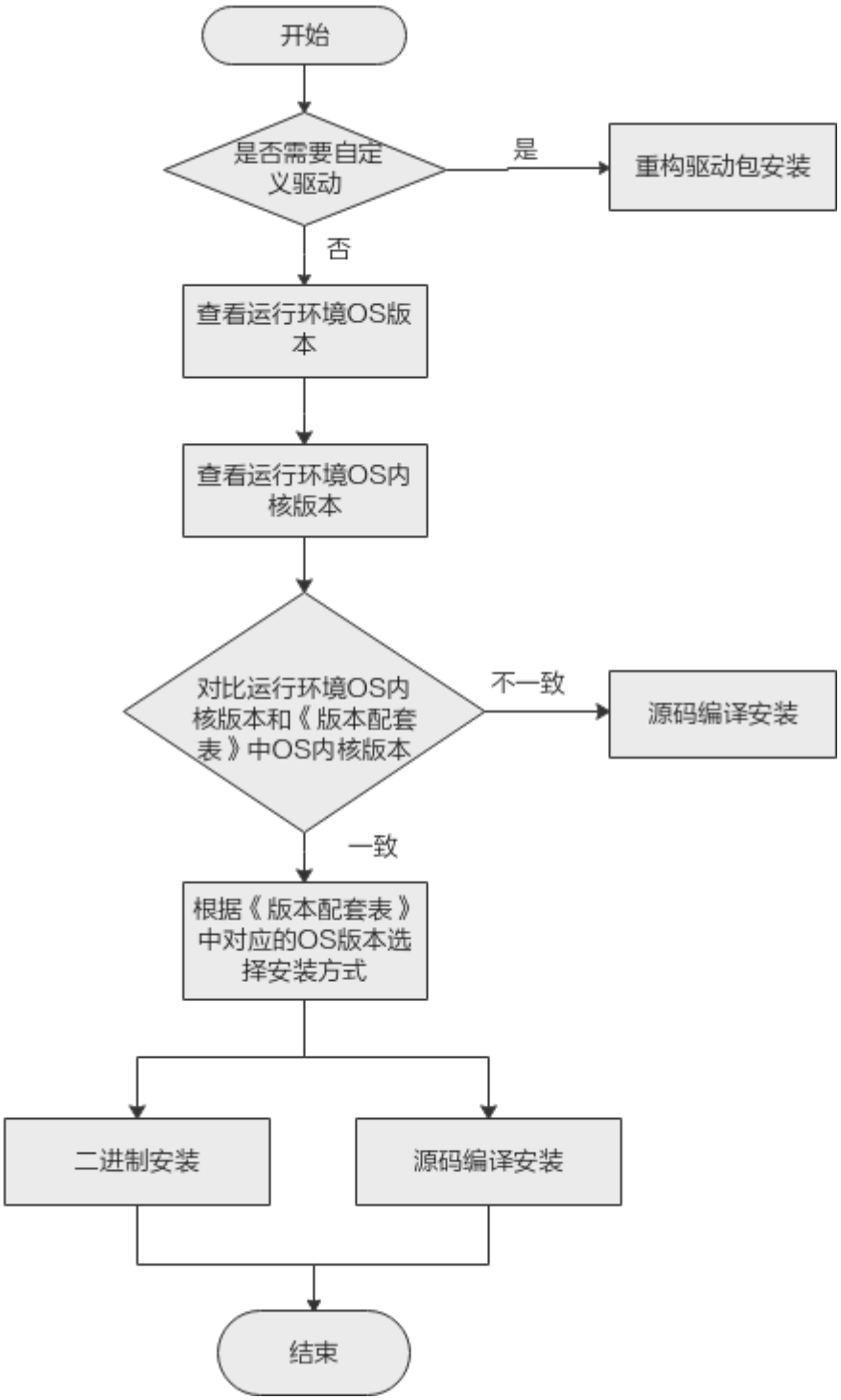
如果设备已经纳管到FusionDirector，安装驱动后，FusionDirector界面上显示的驱动版本会延迟刷新，最长为1个小时。重启设备后版本会立即刷新。

8.2.5.1 确认安装方式

根据**图8-2**选择驱动安装方式。

版权所有 © 华为技术有限公司

图 8-2 确认安装方式流程



说明

- 如果运行环境的操作系统内核版本是《版本配套表》中内核版本的衍生版本，驱动也可以通过源码编译方式安装，但可能会存在无法正常编译的问题。
- 对于EulerOS-Virtual-V200R009C10操作系统，当前host操作系统的内核版本必须与《版本配套表》中要求一致，不支持源码编译。
- EulerOS 2.8场景下，操作系统内核版本必须与《版本配套表》中默认系统版本一致；如果不一致，只能通过升级内核版本的方式解决。否则会导致安装过程报错：部分芯片无法启动。

系统兼容性要求

- 支持PCIe MSI-X中断，每个Atlas 300I 推理卡至少可分配464个MSI-X中断。
- BAR空间大小要求：

  Atlas 300I 推理卡需要三个BAR，空间大小分别为512KB，64MB，256MB。

环境检查

表 8-4 环境检查

序号	检查项	检查命令	检查说明
1	操作系统内核版本	**uname -r**	如果Host系统中安装过run包，内核升级后会出现run包启动异常的问题。出现该问题时请参见**8.4.1 升级系统内核版本时未卸载run包导致run包不可用**解决。
2	GCC版本	**gcc -v**	• 若使用二进制文件直接安装驱动包，安装无须检查GCC版本。 • 若使用源码编译安装驱动包，GCC版本须不低于《版本配套表》中要求的版本。
3	查询系统是否安装过软件包	**lsmod| grep drv_pcie_host**	• 如无内容表示未安装过软件包。可以直接安装软件包。 • 如有内容，表示安装过软件包。需要先卸载驱动包后，再安装新版本软件包。卸载驱动包具体请参见**8.2.7 卸载驱动**。

序号	检查项	检查命令	检查说明
4	检测卡是否正常在位	**lspci | grep d100**	• 如果服务器上有*N*（*N*>0）张Atlas 300I 推理卡（型号 3000），回显中含“d100”字段的行数为4*N*，则表示Atlas 300I 推理卡（型号 3000）正常在位。 88:00.0 Processing accelerators: Huawei Technologies Co., Ltd. Device d100 (rev 20) 89:00.0 Processing accelerators: Huawei Technologies Co., Ltd. Device d100 (rev 20) 8a:00.0 Processing accelerators: Huawei Technologies Co., Ltd. Device d100 (rev 20) 8b:00.0 Processing accelerators: Huawei Technologies Co., Ltd. Device d100 (rev 20) ..... • 如果Atlas 300I 推理卡（型号 3000）所在的服务器是华为服务器，可通过登录iBMC WebUI界面，进入“信息管理 > 系统信息 ”，单击“其他”，Atlas 300I 推理卡（型号 3000）的PCIe卡信息在“PCIe卡”列表中，表示Atlas 300I 推理卡（型号 3000）正常在位。
5（可选）	驱动安装过程中所需的Linux工具	NA	请根据表8-8检查所需的Linux工具并进行安装。如系统中缺失，安装过程中也会有回显信息提示，可以根据提示信息进行安装。
6（可选）	相关配置文件	NA	• 对于**Linx 6.0.90**操作系统，需要检查“/etc/pam.d/su”配置文件的“auth sufficient pam_rootok.so”字段是否被注释掉，若被注释，请参见**8.4.3 root用户切换到普通用户需要输入密码权限**解决。 • 对于**SUSE12SP5**操作系统，需要检查配置文件“10-unsupported-modules.conf”中“allow_unsupported_modules ”的值是否为1。执行**cat /etc/modprobe.d/10-unsupported-modules.conf**命令，查看modprobe限制配置文件中“allow_unsupported_modules ”的值是否“1”。如果不是，使用**vi /etc/modprobe.d/10-unsupported-modules.conf**命令修改“allow_unsupported_modules ”的值为“1”。 **说明** 在**SUSE12SP5**操作系统下，可通过“10-unsupported-modules.conf”配置文件控制非系统自带驱动在系统启动过程中的加载。默认情况下，“10-unsupported-modules.conf”配置文件中“allow_unsupported_modules ”的值为“0”，表示不允许系统启动过程中加载非系统自带驱动；“allow_unsupported_modules ”的值设置为“1”时，将允许系统启动过程中加载非系统自带驱动。

版权所有 © 华为技术有限公司

8.2.5.2 二进制文件直接安装

8.2.5.2.1 安装驱动（适用于.run 格式）

本章节介绍宿主机上的安装方法，以*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run包为例说明，具体操作以客户实际host系统对应驱动包为准。

注意事项

- 如果有安装NPU 1.X.X系列版本的驱动包，在安装NPU 20.X.X及以上系列版本的驱动包前需要先卸载低版本驱动。
- 在软件包安装或升级过程中，不允许对Host或Device进行复位或下电操作，否则会出现Device无法正常启动或者升级失败。

操作步骤

须知

首次安装请按照“**驱动->固件**”的顺序；覆盖安装或升级请按照“**固件->驱动**”的顺序，分别安装软件包。固件安装请参见**8.2.6.1 安装固件（适用于.run格式）**。

首次安装过程中如果使用了--install for all参数，覆盖安装中也需要保留--install for all参数。

对于Atlas 300I 推理卡（型号 3000），为保证驱动版本、固件版本和MCU版本三者保持配套关系，请参考对应版本《**Atlas 300I 推理卡 NPU驱动和固件升级指南（型号 3000, 3010）**》中“升级MCU”章节升级对应的MCU组件。

步骤1 将**8.2.3 获取软件包**准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTY登录服务器的OS命令行。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 执行如下命令，进入软件包所在路径（如“/opt”）。

cd */opt*

步骤5 执行如下命令，增加软件包的可执行权限。

chmod +x *{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run

步骤6 执行如下命令，校验run安装包的一致性和完整性。

./*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run **--check**

若出现如下回显信息，表示软件包校验成功。

```
Verifying archive integrity...  100%   SHA256 checksums are OK. All good.
```

说明

软件包校验使用SHA256加密方式；校验过程中若提示：./*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run does not contain an embedded MD5 checksum.和./*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run does not contain a CRC checksum，表示没有使用MD5和CRC加密方式，可忽略。

步骤7 执行**./***{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run **--full**命令，完成驱动安装，软件包默认安装路径为“/usr/local/Ascend”。

说明

- 安装详细日志路径：/var/log/ascend_seclog/ascend_install.log。
- 安装后软件包的安装路径、安装命令以及运行用户信息记录路径为“/etc/ascend_install.info”。
- 安装run包驱动时，会将动态库libdcmi.so和头文件dcmi_interface_api.h拷贝到“/usr/local/dcmi/”目录下。
- 若创建运行用户是非HwHiAiUser，安装驱动软件包时需要指定运行用户（通过**--install-username=***username* **--install-usergroup=***usergroup*参数指定）。
- 如果指定root用户为运行用户，则需要与--install-for-all参数配合使用，如下所示，该场景下权限控制可能存在安全风险。
  ```
  --install-username=root  --install-usergroup=root  --install-for-all
  ```
- 如果服务器无BMC时，只能采用默认路径安装驱动。如果采用指定路径安装，ipmi_si驱动加载时会出现阻塞，导致磁盘挂载慢，影响npu正常运行。
- Device侧系统类日志通过msnpureport工具传输到Host侧进行查看。导出操作以及导出日志的存储路径具体请参见对应产品和版本的《黑匣子日志参考》的“msnpureport工具使用”章节。容器内不支持查看Device侧系统类日志，也不支持通过msnpureport工具导出Device侧系统类日志。
- 安装驱动过程中出现：“driver ko vermagic(*) is different from the os(*), need rebuild driver”，请参见**8.2.5.3 源码编译安装**解决。

若用户需要指定安装路径，如以“/test/HiAI/”为例。

可执行**./***{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run **--full --install-path=**/test/HiAI/命令，完成安装。

- 若指定路径不存在，则安装时会自动创建目录，若有多层目录，则只有最后一层目录不存在时会自动创建。
- 若指定路径已存在：
  - 若该路径下所有层级目录属主为root用户，则请确保所有层级目录权限至少为755。若不满足要求，请修改路径权限：**chmod 755 路径**
  - 若该路径下有一层目录属主为非root用户，则请用户自行修改为root属主，并确保所有层级目录权限为755。若不满足要求，请修改路径属主为root：**chown root:群组名 路径**

若系统出现如下关键回显信息，则表示驱动安装成功。
```
Driver package installed successfully!
```

步骤8 根据系统提示信息决定是否重启系统，若需要重启，请执行以下命令；否则，请跳过此步骤。

reboot

步骤9 执行**npu-smi info**查看驱动加载是否成功。

若出现如下回显信息，说明加载成功。否则，说明加载失败。请联系华为技术支持处理。

```
+------------------------------------------------------------------------------------------------+
| npu-smi 21.0.4                   Version: 21.0.4.B030                                          |
+-------------------+-----------------+-------------------------------------------------------+
| NPU     Name      | Health          | Power(W)          Temp(C)             Hugepages-Usage(page) |
| Chip    Device    | Bus-Id          | AICore(%)         Memory-Usage(MB)                          |
+===================+=================
+===============================================================+
```

```
| 6      310      | OK            | 12.8            73          0       / 970            |
| 0      3        | 0000:89:00.0  | 0               2703 / 8192                          |
+-------------------+----------------+-----------------------------------------------------------+
| 6      310      | OK            | 12.8            74          0       / 970            |
| 1      4        | 0000:8A:00.0  | 0               2867 / 8192                          |
+-------------------+----------------+-----------------------------------------------------------+
| 6      310      | OK            | 12.8            70          0       / 970            |
| 2      5        | 0000:8B:00.0  | 0               2867 / 8192                          |
+-------------------+----------------+-----------------------------------------------------------+
| 6      310      | OK            | 12.8            65          0       / 970            |
| 3      6        | 0000:8C:00.0  | 0               2867 / 8192                          |
+===================+=================
+=============================================================+
```

说明

- 加载失败，可以执行**dmesg**命令查看linux启动日志。如果出现“*/安装路径*/driver/device/davinci_mini.fd copy err”信息，请参考**8.2.7.2 卸载驱动（适用于.run格式）**章节卸载驱动，并使用默认路径重新进行驱动安装。
- 回显中“npu-smi”后面的字段为npu-smi工具版本号；“Version:”后面的字段为驱动版本号。

----结束

8.2.5.2.2 安装驱动（适用于.rpm 格式）

本章以*{product name}*-npu-driver-*x.x.x*-*<release>*.*{arch}*.rpm为例介绍.rpm格式驱动包安装相关指导，具体请根据实际情况进行替换。

注意事项

- 使用NPU 20.X.X及以上系列版本的rpm软件包安装时，npu-smi工具集成在驱动包内，安装驱动过程中会自动安装npu-smi工具。
- 当前rpm包安装不支持修改安装路径功能，请按照正常rpm流程安装，不支持rpm包的--force命令安装升级。
- Ascend 310内部部分功能，只支持HwHiAiUser用户访问和执行，用户首次安装时系统会自动创建HwHiAiUser用户。
- 如果已安装NPU 1.X.X系列版本的rpm包，在安装NPU 20.X.X及以上系列版本的rpm包前需要先卸载原版本再安装。
- 在软件包安装或升级过程中，不允许对Host或Device进行复位或下电操作，否则会出现Device无法正常启动或者升级失败。

操作步骤

须知

请按照“**驱动->固件**”的顺序，分别安装软件包。固件安装请参见**8.2.6.2 安装固件（适用于.rpm格式）**。

步骤1 将**8.2.3 获取软件包**准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTY登录服务器的OS命令行。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 在软件包所在路径执行如下命令完成Host侧驱动包安装。

rpm -ivh *{product name}-npu-driver-x.x.x-<release>.{arch}.rpm*

说明

软件包安装过程中，若无HwHiAiUser用户系统会自动创建该用户。

若系统出现如下关键回显信息，则表示驱动安装成功。

```
Driver package installed successfully!
```

步骤5 根据系统提示信息决定是否重启系统，若需要重启，请执行以下命令；否则，请跳过此步骤。

reboot

步骤6 执行**npu-smi info**查看驱动加载是否成功。

若出现如下回显信息，说明加载成功。否则，说明加载失败。请联系华为技术支持处理。

```
+------------------------------------------------------------------------------------------------+
| npu-smi 21.0.4                    Version: 21.0.4.B030                                         |
+-------------------+-----------------+-------------------------------------------------------+
| NPU     Name      | Health          | Power(W)          Temp(C)             Hugepages-Usage(page) |
| Chip    Device    | Bus-Id          | AICore(%)         Memory-Usage(MB)                          |
+===================+=================
+=========================================================+
| 6       310       | OK              | 12.8              73            0       / 970              |
| 0       3         | 0000:89:00.0    | 0                 2703 / 8192                            |
+-------------------+-----------------+-------------------------------------------------------+
| 6       310       | OK              | 12.8              74            0       / 970              |
| 1       4         | 0000:8A:00.0    | 0                 2867 / 8192                            |
+-------------------+-----------------+-------------------------------------------------------+
| 6       310       | OK              | 12.8              70            0       / 970              |
| 2       5         | 0000:8B:00.0    | 0                 2867 / 8192                            |
+-------------------+-----------------+-------------------------------------------------------+
| 6       310       | OK              | 12.8              65            0       / 970              |
| 3       6         | 0000:8C:00.0    | 0                 2867 / 8192                            |
+===================+=================
+=========================================================+
```

说明

- 加载失败，可以执行**dmesg**命令查看linux启动日志。如果出现“*/安装路径*/driver/device/davinci_mini.fd copy err” 信息，请参考**8.2.7.2 卸载驱动（适用于.run格式）**章节卸载驱动，并使用默认路径重新进行驱动安装。
- 回显中“npu-smi”后面的字段为npu-smi工具版本号；“Version:”后面的字段为驱动版本号。

----结束

8.2.5.2.3 安装驱动（适用于.deb 格式）

通过 dpkg 安装

使用默认运行用户HwHiAiUser安装

步骤1 将**8.2.3 获取软件包**准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTY登录服务器的OS命令行。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 执行如下命令，进入软件包所在路径（如“/opt”）。

cd /*opt*

步骤5 （**可选**）执行如下命令，调用--install-for-all参数，允许其他用户具有安装群组权限。

export ASCEND_INSTALL_FOR_ALL=true

说明

首次安装过程中如果使用了--install for all参数，覆盖安装中也需要保留--install for all参数。

当安装或者升级携带该参数时，软件包中创建的目录及文件，其他用户权限=安装群组权限。使用该参数将会存在安全风险：其他所有用户都有权限访问安装目录，请谨慎使用。

步骤6 执行如下命令，安装deb驱动包（*驱动包名称请根据实际情况替换*）。

dpkg -i *{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.deb

若系统出现如下关键回显信息，则表示驱动安装成功。

```
Driver package installed successfully!
```

步骤7 根据系统提示信息决定是否重启系统，若需要重启，请执行以下命令；否则，请跳过此步骤。

reboot

步骤8 执行如下命令，查询驱动安装信息。

dpkg -l | grep Ascend

----结束

指定运行用户安装

步骤1 将**8.2.3 获取软件包**准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTY登录服务器的OS命令行。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 执行如下命令，进入软件包所在路径（如“/opt”）。

cd /*opt*

步骤5 （**可选**）执行如下命令，调用--install-for-all参数，允许其他用户具有安装群组权限。

export ASCEND_INSTALL_FOR_ALL=true

说明

首次安装过程中如果使用了--install for all参数，覆盖安装中也需要保留--install for all参数。

当安装或者升级携带该参数时，软件包中创建的目录及文件，其他用户权限=安装群组权限。使用该参数将会存在安全风险：其他所有用户都有权限访问安装目录，请谨慎使用。

版权所有 © 华为技术有限公司

步骤6 设置环境变量。

说明

如果指定环境变量为root用户，则需要与export ASCEND_INSTALL_FOR_ALL=true配合使用，该场景下权限控制可能存在安全风险。

1. 通过export方式设置环境变量，该种方式设置的环境变量只在当前窗口有效，设置完后立即生效。

   export ASCEND_USER_GROUP=*usergroup*

   export ASCEND_USER_NAME=*username*

2. 通过~/.bashrc文件方式设置永久环境变量，操作如下：
   a. 以安装用户在任意目录下执行**vi ~/.bashrc**命令，打开.bashrc文件，在文件最后一行后面添加上述内容。
   b. 执行**:wq!**命令保存文件并退出。
   c. 执行**source ~/.bashrc**命令使其立即生效。

步骤7 将创建的用户名加入创建的用户组。

groupadd ${ASCEND_USER_GROUP}

useradd -g ${ASCEND_USER_GROUP} -d /home/${ASCEND_USER_NAME} -m ${ASCEND_USER_NAME} -s /bin/bash

步骤8 执行如下命令，安装deb驱动包（*驱动包名称请根据实际情况替换*）。

dpkg -i *{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.deb

若系统出现如下关键回显信息，则表示驱动安装成功。

```
Driver package installed successfully!
```

步骤9 根据系统提示信息决定是否重启系统，若需要重启，请执行以下命令；否则，请跳过此步骤。

reboot

步骤10 执行如下命令，查询驱动安装信息。

dpkg -l | grep Ascend

步骤11 （**可选**）执行如下命令，清除相关环境变量。

unset ASCEND_USER_NAME

unset ASCEND_USER_GROUP

unset ASCEND_INSTALL_FOR_ALL

----结束

通过 apt-get 安装

默认用户安装

步骤1 （**可选**）执行如下命令，调用--install-for-all参数，允许其他用户具有安装群组权限。

export ASCEND_INSTALL_FOR_ALL=true

版权所有 © 华为技术有限公司

说明

首次安装过程中如果使用了--install for all参数，覆盖安装中也需要保留--install for all参数。

当安装或者升级携带该参数时，软件包中创建的目录及文件，其他用户权限=安装群组权限。使用该参数将会存在安全风险：其他所有用户都有权限访问安装目录，请谨慎使用。

步骤2 执行以下命令，安装本地源中的deb驱动包。

- **首次安装**

  apt-get install ascend310-driver

- **重新安装**

  apt-get --reinstall install ascend310-driver

  若系统出现如下关键回显信息，则表示驱动安装成功。

```
Driver package installed successfully!
```

步骤3 根据系统提示信息决定是否重启系统，若需要重启，请执行以下命令；否则，请跳过此步骤。

reboot

步骤4 执行如下命令，查询驱动安装信息。

apt list ascend310-driver

----结束

指定用户安装

步骤1 （**可选**）执行如下命令，调用--install-for-all参数，允许其他用户具有安装群组权限。

export ASCEND_INSTALL_FOR_ALL=true

说明

首次安装过程中如果使用了--install for all参数，覆盖安装中也需要保留--install for all参数。

当安装或者升级携带该参数时，软件包中创建的目录及文件，其他用户权限=安装群组权限。使用该参数将会存在安全风险：其他所有用户都有权限访问安装目录，请谨慎使用。

步骤2 设置环境变量。

说明

如果指定环境变量为root用户，则需要与export ASCEND_INSTALL_FOR_ALL=true配合使用，该场景下权限控制可能存在安全风险。

1. 通过export方式设置环境变量，该种方式设置的环境变量只在当前窗口有效，设置完后立即生效。

   export ASCEND_USER_GROUP=*usergroup*

   export ASCEND_USER_NAME=*username*

2. 通过~/.bashrc文件方式设置永久环境变量，操作如下：

   a. 以安装用户在任意目录下执行**vi ~/.bashrc**命令，打开.bashrc文件，在文件最后一行后面添加上述内容。

   b. 执行**:wq!**命令保存文件并退出。

   c. 执行**source ~/.bashrc**命令使其立即生效。

步骤3 将创建的用户名加入创建的用户组。

groupadd ${ASCEND_USER_GROUP}

useradd -g ${ASCEND_USER_GROUP} -d /home/${ASCEND_USER_NAME} -m ${ASCEND_USER_NAME} -s /bin/bash

步骤4 执行以下命令，安装本地源中的deb驱动包。

- **首次安装**

  apt-get install ascend310-driver

- **重新安装**

  apt-get --reinstall install ascend310-driver

  若系统出现如下关键回显信息，则表示驱动安装成功。

  ```
  Driver package installed successfully!
  ```

步骤5 根据系统提示信息决定是否重启系统，若需要重启，请执行以下命令；否则，请跳过此步骤。

reboot

步骤6 执行如下命令，查询驱动安装信息。

apt list ascend310-driver

步骤7 （**可选**）执行如下命令，清除相关环境变量。

unset ASCEND_USER_NAME

unset ASCEND_USER_GROUP

unset ASCEND_INSTALL_FOR_ALL

----结束

8.2.5.3 源码编译安装

当用户当前操作系统内核版本更新或与默认不一致，可以基于安装包驱动源码重新构建并安装使用，该场景下不会触发内核版本升级，会基于用户当前内核版本编译出相应的内核驱动。软件包在安装时会自动触发驱动源码编译，编译完成后自动进行软件包安装。

如果使用该功能，则需要提前检查并安装相关依赖工具。具体操作请切换到root用户执行。

8.2.5.3.1 安装依赖

make 工具

执行**make -v**命令，若能查询到make工具的版本，则make已安装。

默认内核源码路径或 dkms、gcc、linux-header 等安装依赖

- 对于以下各操作系统，请执行如下命令检查dkms、gcc、linux-header等安装依赖是否存在。
- 对于其他操作系统，检查默认内核源码路径是否存在。例如：/lib/modules/`uname -r`/build

  执行**ls /lib/modules/`uname -r`/build**命令，查看路径是否存在。

- 若存在，安装驱动包时则会自动使用内核进行驱动编译。
- 若不存在，您可以在安装时提供对应的源码路径，具体请参见步骤8。

说明

以下操作系统中的DKMS软件包建议使用dkms-2.6.1-1.el7.noarch.rpm。

Ubuntu 18.04 操作系统

需要安装dkms、gcc、linux-header软件包。执行如下命令检查：

dpkg-query -s dkms

dpkg-query -s gcc

dpkg-query -s linux-headers-$(uname -r)

若未安装请自行在相应操作系统厂商网站获取并安装。

Ubuntu20.04 操作系统

需要安装dkms、gcc、linux-header软件包。执行如下命令检查：

dpkg-query -s dkms

dpkg-query -s gcc

dpkg-query -s linux-headers-$(uname -r)

若未安装请自行在相应操作系统厂商网站获取并安装

CentOS 7.6 操作系统

需要安装dkms、gcc、kernel-headers、kernel-devel软件包。执行如下命令检查：

rpm -qa | grep dkms

rpm -qa | grep gcc

rpm -qa | grep kernel-headers-$(uname -r)

rpm -qa | grep kernel-devel-$(uname -r)

若未安装请自行在相应操作系统厂商网站获取并安装。

CentOS 8.2 操作系统

需要安装dkms、gcc、kernel-headers、kernel-devel软件包。执行如下命令检查：

rpm -qa | grep dkms

rpm -qa | grep gcc

rpm -qa | grep kernel-headers-$(uname -r)

rpm -qa | grep kernel-devel-$(uname -r)

若未安装请自行在相应操作系统厂商网站获取并安装。

EulerOS 2.8 操作系统

需要安装gcc、kernel-headers、kernel-devel软件包。执行如下命令检查：

rpm -qa | grep gcc

rpm -qa | grep kernel-headers-$(uname -r)

rpm -qa | grep kernel-devel-$(uname -r)

若未安装请自行在相应操作系统厂商网站获取并安装。

EulerOS 2.10 操作系统

需要安装gcc、kernel-headers、kernel-devel软件包。执行如下命令检查：

rpm -qa | grep gcc

rpm -qa | grep kernel-headers-$(uname -r)

rpm -qa | grep kernel-devel-$(uname -r)

若未安装请自行在相应操作系统厂商网站获取并安装。

SUSE12SP5 操作系统

需要安装gcc、kernel-default、kernel-default-devel软件包。执行如下命令检查：

rpm -qa | grep gcc

rpm -qa | grep kernel-default

rpm -qa | grep kernel-default-devel

若未安装请自行在相应操作系统厂商网站获取并安装。

EulerOS 2.9 操作系统

需要安装kernel-headers、gcc 7.3.0、elfutils-devel、make等软件包。操作如下：

1. 请获取对应版本OS的“devel_tools.tar.gz”工具组件。
2. 上传“devel_tools.tar.gz”到运行环境的任一目录。如：/opt下。
3. 进入“devel_tools.tar.gz”所在路径并进行解压。

   cd /opt

   tar -xzvf devel_tools.tar.gz
4. 在3的解压目录下，执行以下命令安装kernel-headers、gcc 7.3.0、elfutils-devel、make等软件包。
   - 对于**EulerOS 2.9+aarch64**系统，请执行如下命令：

     rpm -ivh kernel-headers-4.19.90-vhulk2009.2.0.h269.eulerosv2r9.aarch64.rpm kernel-devel-4.19.90-vhulk2009.2.0.h269.eulerosv2r9.aarch64.rpm

     rpm -ivh binutils-2.34-1.h11.eulerosv2r9.aarch64.rpm binutils-extra-2.34-1.h11.eulerosv2r9.aarch64.rpm binutils-devel-2.34-1.h11.eulerosv2r9.aarch64.rpm zlib-devel-1.2.11-17.eulerosv2r9.aarch64.rpm cpp-7.3.0-20190804.h39.eulerosv2r9.aarch64.rpm

libmpc-1.1.0-4.eulerosv2r9.aarch64.rpm glibc-devel-2.28-59.eulerosv2r9.aarch64.rpm libxcrypt-devel-4.4.8-4.h2.eulerosv2r9.aarch64.rpm

rpm -ivh gcc-7.3.0-20190804.h39.eulerosv2r9.aarch64.rpm

rpm -ivh elfutils-devel-0.179-1.h3.eulerosv2r9.aarch64.rpm

rpm -ivh make-4.2.1-15.eulerosv2r9.aarch64.rpm

Tlinux 2.4 操作系统

由于内核进行过针对性适配，需要使用版本为8.2.1的GCC编译器，因此编译驱动前需保证8.2.1的GCC编译器已经部署。操作步骤如下：

1. 获取GCC 8.2.1版本编译器的安装包。软件包获取网址：**https://mirrors.tencent.com/tlinux/2.4/arm64/tlinux-sclo/aarch64/tl/devtoolset-8/devtoolset-8-gcc-8.2.1-3.tl2.aarch64.rpm**。
2. 登录运行环境。
3. 上传gcc编译器的安装包到指定路径下，如“/tmp”。
4. 执行以下命令安装GCC编译器。

   rpm -Uvh devtoolset-8-gcc-8.2.1-3.tl2.aarch64.rpm --force --nodeps
5. 安装完成后，执行以下命令查找GCC 8.2.1的可以执行文件所在路径。

   find / -name gcc

   执行文件路径如下红框所示：

   ```
   /opt/rh/devtoolset-8/root/usr/libexec/gcc
   /opt/rh/devtoolset-8/root/usr/bin/gcc
   /opt/rh/devtoolset-8/root/usr/lib/gcc
   ```
6. 配置环境变量。

   export PATH=/opt/rh/devtoolset-8/root/usr/bin/:$PATH
7. 执行以下命令，验证GCC编译器是否安装成功。

   gcc --version

   如出现以下回显，安装后的gcc版本和目标版本一致，则表示安装成功。

   ```
   [root@TENCENT64 bin]# gcc --version
   gcc (GCC) 8.2.1 20180905 (Red Hat 8.2.1-3)
   Copyright (C) 2018 Free Software Foundation, Inc.
   This is free software; see the source for copying conditions.  There is NO
   warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
   ```

8.2.5.3.2 安装驱动（适用于.run 格式）

本章节介绍宿主机上的安装方法，以*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run包为例说明，具体操作以客户实际host系统对应驱动包为准。

前提条件

驱动包安装前需要检查并安装相关依赖，具体请参见**8.2.5.3.1 安装依赖**。

注意事项

- 如果有安装NPU 1.X.X系列版本的驱动包，在安装NPU 20.X.X及以上系列版本的驱动包前需要先卸载低版本驱动。

版权所有 © 华为技术有限公司

- 在软件包安装或升级过程中，不允许对Host或Device进行复位或下电操作，否则会出现Device无法正常启动或者升级失败。

操作步骤

须知

首次安装请按照“**驱动->固件**”的顺序；覆盖安装或升级请按照“**固件->驱动**”的顺序，分别安装软件包。固件安装请参见**8.2.6.1 安装固件（适用于.run格式）**。

首次安装过程中如果使用了--install for all参数，覆盖安装中也需要保留--install for all参数。

对于Atlas 300I 推理卡（型号 3000），为保证驱动版本、固件版本和MCU版本三者保持配套关系，请参考对应版本《**Atlas 300I 推理卡 NPU驱动和固件升级指南（型号 3000, 3010）**》中“升级MCU”章节升级对应的MCU组件。

步骤1 将**8.2.3 获取软件包**准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTY登录服务器的OS命令行。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 执行如下命令，进入软件包所在路径（如“/opt”）。

cd /*opt*

步骤5 执行如下命令，增加软件包的可执行权限。

chmod +x *{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run

步骤6 执行如下命令，校验run安装包的一致性和完整性。

./*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run **--check**

若出现如下回显信息，表示软件包校验成功。

```
Verifying archive integrity...  100%   SHA256 checksums are OK. All good.
```

说明

软件包校验使用SHA256加密方式；校验过程中若提示：./*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run does not contain an embedded MD5 checksum.和./*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run does not contain a CRC checksum，表示没有使用MD5和CRC加密方式，可忽略。

步骤7 执行**./***{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run **--full**命令，完成驱动安装，软件包默认安装路径为“/usr/local/Ascend”。

说明

- 安装详细日志路径：/var/log/ascend_seclog/ascend_install.log。
- 安装后软件包的安装路径、安装命令以及运行用户信息记录路径为“/etc/ascend_install.info”。
- 安装run包驱动时，会将动态库libdcmi.so和头文件dcmi_interface_api.h拷贝到“/usr/local/dcmi/”目录下。
- 若创建运行用户是非HwHiAiUser，安装驱动软件包时需要指定运行用户（通过**--install-username=***username* **--install-usergroup=***usergroup*参数指定）。
- 如果指定root用户为运行用户，则需要与--install-for-all参数配合使用，如下所示，该场景下权限控制可能存在安全风险。
  ```
  --install-username=root  --install-usergroup=root  --install-for-all
  ```
- 如果服务器无BMC时，只能采用默认路径安装驱动。如果采用指定路径安装，ipmi_si驱动加载时会出现阻塞，导致磁盘挂载慢，影响npu正常运行。
- Device侧系统类日志通过msnpureport工具传输到Host侧进行查看。导出操作以及导出日志的存储路径具体请参见对应产品和版本的《黑匣子日志参考》的“msnpureport工具使用”章节。容器内不支持查看Device侧系统类日志，也不支持通过msnpureport工具导出Device侧系统类日志。

若用户需要指定安装路径，如以“/test/HiAI/”为例。

可执行**./***{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run **--full --install-path=**/test/HiAI/命令，完成安装。

- 若指定路径不存在，则安装时会自动创建目录，若有多层目录，则只有最后一层目录不存在时会自动创建。
- 若指定路径已存在：
  - 若该路径下所有层级目录属主为root用户，则请确保所有层级目录权限至少为755。若不满足要求，请修改路径权限：**chmod 755 路径**
  - 若该路径下有一层目录属主为非root用户，则请用户自行修改为root属主，并确保所有层级目录权限为755。若不满足要求，请修改路径属主为root：**chown root:群组名 路径**

步骤8 （**可选**）安装时若出现以下回显，说明未安装dkms，默认内核源码路径如“/lib/modules/`uname -r`/build”不存在。请根据系统提示输入相应内容：

```
[WARNING]rebuild ko has something wrong, detail in /var/log/ascend_seclog/ascend_rebuild.log
Do you want to try build driver after input kernel absolute path? [y/n]:
```

若需要继续安装，请输入**y**。

出现如下回显后，请根据提示输入您提供的内核源码的实际路径，如：**/lib/modules/`uname -r`/build-bak**。

```
Please input your kernel absolute path:
```

按“Enter”键继续安装。

说明

- 若已安装dkms及相应kernel-header、kernel-devel等组件，则会自动执行dkms驱动编译安装。
- 若未安装dkms，但默认内核源码路径如“/lib/modules/`uname -r`/build”已存在，则会自动使用该内核进行驱动编译。

步骤9 若系统出现如下关键回显信息，则表示驱动安装成功。

```
Driver package installed successfully!
```

步骤10 根据系统提示信息决定是否重启系统，若需要重启，请执行以下命令；否则，请跳过此步骤。

reboot

步骤11 执行**npu-smi info**查看驱动加载是否成功。

若出现如下回显信息，说明加载成功。否则，说明加载失败。请联系华为技术支持处理。

```
+------------------------------------------------------------------------------------------------+
| npu-smi 21.0.4                    Version: 21.0.4.B030                                         |
+-------------------+-----------------+-------------------------------------------------------+
| NPU     Name      | Health          | Power(W)          Temp(C)            Hugepages-Usage(page) |
| Chip    Device    | Bus-Id          | AICore(%)         Memory-Usage(MB)                         |
+===================+=================
+==============================================================+
| 6       310       | OK              | 12.8              73            0       / 970              |
| 0       3         | 0000:89:00.0    | 0                 2703 / 8192                              |
+-------------------+-----------------+-------------------------------------------------------+
| 6       310       | OK              | 12.8              74            0       / 970              |
| 1       4         | 0000:8A:00.0    | 0                 2867 / 8192                              |
+-------------------+-----------------+-------------------------------------------------------+
| 6       310       | OK              | 12.8              70            0       / 970              |
| 2       5         | 0000:8B:00.0    | 0                 2867 / 8192                              |
+-------------------+-----------------+-------------------------------------------------------+
| 6       310       | OK              | 12.8              65            0       / 970              |
| 3       6         | 0000:8C:00.0    | 0                 2867 / 8192                              |
+===================+=================
+==============================================================+
```

说明

- 加载失败，可以执行**dmesg**命令查看linux启动日志。如果出现“*/安装路径*/driver/device/davinci_mini.fd copy err”信息，请参考**8.2.7.2 卸载驱动（适用于.run格式）**章节卸载驱动，并使用默认路径重新进行驱动安装。
- 回显中“npu-smi”后面的字段为npu-smi工具版本号；“Version:”后面的字段为驱动版本号。

----结束

8.2.5.3.3 安装驱动（适用于.rpm 格式）

本章以*{product name}*-npu-driver-*x.x.x-<release>.{arch}*.rpm为例介绍.rpm格式驱动包安装相关指导，具体请根据实际情况进行替换。

前提条件

驱动包安装前需要检查并安装相关依赖，具体请参见**8.2.5.3.1 安装依赖**。

注意事项

- 使用NPU 20.X.X及以上系列版本的rpm软件包安装时，npu-smi工具集成在驱动包内，安装驱动过程中会自动安装npu-smi工具。
- 当前rpm包安装不支持修改安装路径功能，请按照正常rpm流程安装，不支持rpm包的--force命令安装升级。
- Ascend 310内部部分功能，只支持HwHiAiUser用户访问和执行，用户首次安装时系统会自动创建HwHiAiUser用户。
- 如果已安装NPU 1.X.X系列版本的rpm包，在安装NPU 20.X.X及以上系列版本的rpm包前需要先卸载原版本再安装。

- 在软件包安装或升级过程中，不允许对Host或Device进行复位或下电操作，否则会出现Device无法正常启动或者升级失败。

操作步骤

须知

请按照“**驱动->固件**”的顺序，分别安装软件包。固件安装请参见**8.2.6.2 安装固件（适用于.rpm格式）**。

步骤1 将**8.2.3 获取软件包**准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTY登录服务器的OS命令行。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 在软件包所在路径执行如下命令完成Host侧驱动包安装。

rpm -ivh *{product name}-npu-driver-x.x.x-<release>.{arch}.rpm*

说明

软件包安装过程中，若无HwHiAiUser用户系统会自动创建该用户。

若系统出现如下关键回显信息，则表示驱动安装成功。

```
Driver package installed successfully!
```

步骤5 根据系统提示信息决定是否重启系统，若需要重启，请执行以下命令；否则，请跳过此步骤。

reboot

步骤6 执行**npu-smi info**查看驱动加载是否成功。

若出现如下回显信息，说明加载成功。否则，说明加载失败。请联系华为技术支持处理。

```
+------------------------------------------------------------------------------------------------+
| npu-smi 21.0.4                    Version: 21.0.4.B030                                         |
+-------------------+-----------------+-------------------------------------------------------+
| NPU     Name      | Health          | Power(W)          Temp(C)             Hugepages-Usage(page) |
| Chip    Device    | Bus-Id          | AICore(%)         Memory-Usage(MB)                          |
+===================+=================
+==============================================================+
| 6       310       | OK              | 12.8              73            0       / 970              |
| 0       3         | 0000:89:00.0    | 0                 2703 / 8192                              |
+-------------------+-----------------+-------------------------------------------------------+
| 6       310       | OK              | 12.8              74            0       / 970              |
| 1       4         | 0000:8A:00.0    | 0                 2867 / 8192                              |
+-------------------+-----------------+-------------------------------------------------------+
| 6       310       | OK              | 12.8              70            0       / 970              |
| 2       5         | 0000:8B:00.0    | 0                 2867 / 8192                              |
+-------------------+-----------------+-------------------------------------------------------+
| 6       310       | OK              | 12.8              65            0       / 970              |
| 3       6         | 0000:8C:00.0    | 0                 2867 / 8192                              |
+===================+=================
+==============================================================+
```


版权所有 © 华为技术有限公司

说明

- 加载失败，可以执行**dmesg**命令查看linux启动日志。如果出现“/*安装路径*/driver/device/davinci_mini.fd copy err”信息，请参考**8.2.7.2 卸载驱动（适用于.run格式）**章节卸载驱动，并使用默认路径重新进行驱动安装。
- 回显中“npu-smi”后面的字段为npu-smi工具版本号；“Version:”后面的字段为驱动版本号。

----结束

8.2.5.3.4 安装驱动（适用于.deb 格式）

驱动包安装前需要检查并安装相关依赖，具体请参见**8.2.5.3.1 安装依赖**。

通过 dpkg 安装

使用默认运行用户HwHiAiUser安装

步骤1 将**8.2.3 获取软件包**准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTY登录服务器的OS命令行。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 执行如下命令，进入软件包所在路径（如“/opt”）。

cd /*opt*

步骤5 （**可选**）执行如下命令，调用--install-for-all参数，允许其他用户具有安装群组权限。

export ASCEND_INSTALL_FOR_ALL=true

说明

首次安装过程中如果使用了--install for all参数，覆盖安装中也需要保留--install for all参数。

当安装或者升级携带该参数时，软件包中创建的目录及文件，其他用户权限=安装群组权限。使用该参数将会存在安全风险：其他所有用户都有权限访问安装目录，请谨慎使用。

步骤6 执行如下命令，安装deb驱动包（*驱动包名称请根据实际情况替换*）。

dpkg -i *{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.deb

若系统出现如下关键回显信息，则表示驱动安装成功。

```
Driver package installed successfully!
```

步骤7 根据系统提示信息决定是否重启系统，若需要重启，请执行以下命令；否则，请跳过此步骤。

reboot

步骤8 执行如下命令，查询驱动安装信息。

dpkg -l | grep Ascend

----结束

指定运行用户安装

版权所有 © 华为技术有限公司

步骤1 将8.2.3 获取软件包准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTY登录服务器的OS命令行。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 执行如下命令，进入软件包所在路径（如“/opt”）。

cd */opt*

步骤5 （**可选**）执行如下命令，调用--install-for-all参数，允许其他用户具有安装群组权限。

export ASCEND_INSTALL_FOR_ALL=true

说明

首次安装过程中如果使用了--install for all参数，覆盖安装中也需要保留--install for all参数。当安装或者升级携带该参数时，软件包中创建的目录及文件，其他用户权限=安装群组权限。使用该参数将会存在安全风险：其他所有用户都有权限访问安装目录，请谨慎使用。

步骤6 设置环境变量。

说明

如果指定环境变量为root用户，则需要与export ASCEND_INSTALL_FOR_ALL=true配合使用，该场景下权限控制可能存在安全风险。

1. 通过export方式设置环境变量，该种方式设置的环境变量只在当前窗口有效，设置完后立即生效。

   export ASCEND_USER_GROUP=*usergroup*

   export ASCEND_USER_NAME=*username*

2. 通过~/.bashrc文件方式设置永久环境变量，操作如下：
   a. 以安装用户在任意目录下执行**vi ~/.bashrc**命令，打开.bashrc文件，在文件最后一行后面添加上述内容。
   b. 执行**:wq!**命令保存文件并退出。
   c. 执行**source ~/.bashrc**命令使其立即生效。

步骤7 将创建的用户名加入创建的用户组。

groupadd ${ASCEND_USER_GROUP}

useradd -g ${ASCEND_USER_GROUP} -d /home/${ASCEND_USER_NAME} -m ${ASCEND_USER_NAME} -s /bin/bash

步骤8 执行如下命令，安装deb驱动包（*驱动包名称请根据实际情况替换*）。

dpkg -i *{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.deb

若系统出现如下关键回显信息，则表示驱动安装成功。

```
Driver package installed successfully!
```

步骤9 根据系统提示信息决定是否重启系统，若需要重启，请执行以下命令；否则，请跳过此步骤。

reboot

步骤10 执行如下命令，查询驱动安装信息。

dpkg -l | grep Ascend

步骤11 （可选）执行如下命令，清除相关环境变量。

unset ASCEND_USER_NAME

unset ASCEND_USER_GROUP

unset ASCEND_INSTALL_FOR_ALL

----结束

通过 apt-get 安装

默认用户安装

步骤1 （可选）执行如下命令，调用--install-for-all参数，允许其他用户具有安装群组权限。

export ASCEND_INSTALL_FOR_ALL=true

说明

首次安装过程中如果使用了--install for all参数，覆盖安装中也需要保留--install for all参数。

当安装或者升级携带该参数时，软件包中创建的目录及文件，其他用户权限=安装群组权限。使用该参数将会存在安全风险：其他所有用户都有权限访问安装目录，请谨慎使用。

步骤2 执行以下命令，安装本地源中的deb驱动包。

- **首次安装**

  apt-get install ascend310-driver

- **重新安装**

  apt-get --reinstall install ascend310-driver

  若系统出现如下关键回显信息，则表示驱动安装成功。

```
Driver package installed successfully!
```

步骤3 根据系统提示信息决定是否重启系统，若需要重启，请执行以下命令；否则，请跳过此步骤。

reboot

步骤4 执行如下命令，查询驱动安装信息。

apt list ascend310-driver

----结束

指定用户安装

步骤1 （可选）执行如下命令，调用--install-for-all参数，允许其他用户具有安装群组权限。

export ASCEND_INSTALL_FOR_ALL=true

说明

首次安装过程中如果使用了--install for all参数，覆盖安装中也需要保留--install for all参数。

当安装或者升级携带该参数时，软件包中创建的目录及文件，其他用户权限=安装群组权限。使用该参数将会存在安全风险：其他所有用户都有权限访问安装目录，请谨慎使用。

步骤2 设置环境变量。

说明

如果指定环境变量为root用户，则需要与export ASCEND_INSTALL_FOR_ALL=true配合使用，该场景下权限控制可能存在安全风险。

1. 通过export方式设置环境变量，该种方式设置的环境变量只在当前窗口有效，设置完后立即生效。

   export ASCEND_USER_GROUP=*usergroup*

   export ASCEND_USER_NAME=*username*

2. 通过~/.bashrc文件方式设置永久环境变量，操作如下：

   a. 以安装用户在任意目录下执行**vi ~/.bashrc**命令，打开.bashrc文件，在文件最后一行后面添加上述内容。

   b. 执行**:wq!**命令保存文件并退出。

   c. 执行**source ~/.bashrc**命令使其立即生效。

步骤3 将创建的用户名加入创建的用户组。

groupadd ${ASCEND_USER_GROUP}

useradd -g ${ASCEND_USER_GROUP} -d /home/${ASCEND_USER_NAME} -m ${ASCEND_USER_NAME} -s /bin/bash

步骤4 执行以下命令，安装本地源中的deb驱动包。

- **首次安装**

  apt-get install ascend310-driver

- **重新安装**

  apt-get --reinstall install ascend310-driver

  若系统出现如下关键回显信息，则表示驱动安装成功。

  ```
  Driver package installed successfully!
  ```

步骤5 根据系统提示信息决定是否重启系统，若需要重启，请执行以下命令；否则，请跳过此步骤。

reboot

步骤6 执行如下命令，查询驱动安装信息。

apt list ascend310-driver

步骤7 （**可选**）执行如下命令，清除相关环境变量。

unset ASCEND_USER_NAME

unset ASCEND_USER_GROUP

unset ASCEND_INSTALL_FOR_ALL

----结束

8.2.5.4 重构驱动包安装

8.2.5.4.1 安装依赖

GCC 和内核源码

确认目标OS及其内核版本号对应的GCC及内核源码是否已存在，具体请参见**8.2.5.3.1 安装依赖**。

makeself 2.4.0 工具

下载开源工具：makeself 2.4.0。下载地址请参见：**https://github.com/megastep/makeself**或**https://github.com/megastep/makeself/releases/tag/release-2.4.0**。

pigz 工具

执行如下命令，检查pigz工具是否已安装。例如：

- Ubuntu操作系统：

  dpkg-query -s pigz

- CentOS操作系统：

  rpm -qa |grep pigz

若未安装请使用对应的软件管理器进行安装。例如：

- Ubuntu操作系统：

  apt install pigz

- CentOS操作系统：

  yum install pigz

8.2.5.4.2 重构驱动包

用户可以根据本章节内容，构建一个满足目标OS版本，内核版本，gcc版本要求的驱动安装包。本章节以*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run包为例说明，具体操作以客户实际host系统对应驱动包为准。

重构驱动包目前支持如下几种方式：

- 直接重构驱动包
- 解压run包后重构驱动包
- 使用重构工具重构驱动包

须知

建议用户使用如下两种方式。

- 若不需要更改run包中源码的场景；则直接重构驱动包。具体请参见**直接重构驱动包**。
- 若需要修改驱动源码，则解压run包后重构驱动包。具体请参见**解压后重构驱动包**。

版权所有 © 华为技术有限公司

前提条件

编译前需要安装相关依赖，具体请参见**8.2.5.4.1 安装依赖**。

直接重构驱动包

步骤1 获取通用驱动包，详细信息请参见**8.2.3 获取软件包**。

步骤2 上传通用驱动包到运行环境的任意目录。如：/opt下。

步骤3 执行以下命令，进入**步骤2**软件包所在路径，如/opt下。

cd /*opt*

步骤4 执行以下命令，增加通用驱动包的执行权限。

chmod +x *{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run

步骤5 在root用户下执行以下命令，构建新驱动包。

命令：**./***{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run **--repack** *[package_name]*

示例：**./***{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run **--repack** *{product name}*-npu-driver-repack.run

说明

若package_name参数为空，则会在当前路径下生成一个命名为“原驱动包名称-custom.run”的文件。

步骤6 请参见**8.2.5.4.3 安装驱动（适用于.run格式）**，安装驱动。

----结束

解压后重构驱动包

步骤1 获取通用驱动包，详细信息请参见**8.2.3 获取软件包**。

步骤2 上传通用驱动包到运行环境的任意目录。如：/opt下。

步骤3 执行以下命令，进入**步骤2**软件包所在路径，如/opt下。

cd /*opt*

步骤4 执行以下命令，增加通用驱动包的执行权限。

chmod +x *{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run

步骤5 执行以下命令，解压通用驱动包到指定目录。

命令：**./***{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run **--noexec --extract=***目标路径*

示例：**./***{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run **--noexec --extract=./tmp**

步骤6 在root用户下执行以下命令，构建新驱动包。

命令：**./***{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run **--repack-path=***<path>* *[package_name]*

示例：**./***{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run **--repack-path=tmp/***{product name}*-npu-driver-repack.run

说明

若package_name参数为空，则会在当前路径下生成一个命名为“原驱动包名称-custom.run”的文件。

步骤7 请参见**8.2.5.4.3 安装驱动（适用于.run格式）**，安装驱动。

----结束

使用重构工具重构驱动包

步骤1 根据**8.2.3 获取软件包**获取源码驱动包和重构工具包。

步骤2 上传源码驱动包、重构工具包、开源软件“makeself-release-2.4.0.zip”到运行环境的同一目录。如：/opt下。

步骤3 执行以下命令，进入**步骤2**软件包所在路径，如/opt下。

cd /*opt*

步骤4 执行以下命令，增加源码驱动包的执行权限。

chmod +x *{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run

步骤5 执行以下命令，解压源码驱动包到指定目录。

./*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run **--noexec --extract=./tmp**

步骤6 执行以下命令，解压重构工具包到当前目录。

tar -xzvf *{product name}-npu-driver-repack-tools_x.x.x.tar.gz* **--strip-components 1**

步骤7 执行以下命令，解压软件包“makeself-release-2.4.0.zip”。

unzip makeself-release-2.4.0.zip

步骤8 执行以下命令，构建新驱动包。

命令：**bash build.sh** *makeself.sh路径 包解压的路径 输出包名*

示例：**bash build.sh ./makeself-release-2.4.0/makeself.sh tmp/** *{product name}*-npu-driver-repack.run

步骤9 请参见**8.2.5.4.3 安装驱动（适用于.run格式）**，安装驱动。

----结束

8.2.5.4.3 安装驱动（适用于.run 格式）

本章以**8.2.5.4.2 重构驱动包**的*{product name}*-npu-driver-repack.run为例介绍.run格式驱动包安装相关指导，具体请根据实际情况进行替换。

注意事项

- 如果有安装NPU 1.X.X系列版本的驱动包，在安装NPU 20.X.X及以上系列版本的驱动包前需要先卸载低版本驱动。

版权所有 © 华为技术有限公司

- 在软件包安装或升级过程中，不允许对Host或Device进行复位或下电操作，否则会出现Device无法正常启动或者升级失败。

操作步骤

须知

首次安装请按照“**驱动->固件**”的顺序；覆盖安装或升级请按照“**固件->驱动**”的顺序，分别安装软件包。固件安装请参见**8.2.6.1 安装固件（适用于.run格式）**。

首次安装过程中如果使用了--install for all参数，覆盖安装中也需要保留--install for all参数。

对于Atlas 300I 推理卡（型号 3000），为保证驱动版本、固件版本和MCU版本三者保持配套关系，请参考对应版本《**Atlas 300I 推理卡 NPU驱动和固件升级指南（型号 3000, 3010）**》中“升级MCU”章节升级对应的MCU组件。

步骤1 将**8.2.5.4.2 重构驱动包**构建的驱动包上传至Linux系统任意目录下（如/opt）。

步骤2 使用PuTTY登录服务器的OS命令行。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 执行如下命令，进入软件包所在路径（如“/opt”）。

cd */opt*

步骤5 执行如下命令，校验run安装包的一致性和完整性。

./*{product name}-npu-driver-repack.run* **--check**

出现如下回显信息，表示软件包校验成功。

```
Verifying archive integrity...  100%   SHA256 checksums are OK. All good.
```

说明

软件包校验使用SHA256加密方式；校验过程中若提示：*./{product name}-npu-driver-repack.run* does not contain an embedded MD5 checksum.和*./{product name}-npu-driver-repack.run* does not contain a CRC checksum，表示没有使用MD5和CRC加密方式，可忽略。

步骤6 执行安装命令完成安装。

- 若不指定安装路径，可执行**./***{product name}*-npu-driver-repack.run **--full**命令，完成安装。
- 若用户指定安装路径，以“/test/HiAI/”为例。

  可执行**./***{product name}-npu-driver-repack.run* **--full --install-path=** /test/HiAI/命令，完成安装。

说明

- 安装详细日志路径：/var/log/ascend_seclog/ascend_install.log。
- 安装后软件包的安装路径、安装命令以及运行用户信息记录路径为“/etc/ascend_install.info”。
- 安装run包驱动时，会将动态库libdcmi.so和头文件dcmi_interface_api.h拷贝到“/usr/local/dcmi/”目录下。
- 若创建运行用户是非HwHiAiUser，安装驱动软件包时需要指定运行用户（通过**--install-username=***username* **--install-usergroup=***usergroup*参数指定）。
- 如果指定root用户为运行用户，则需要与--install-for-all参数配合使用，如下所示，该场景下权限控制可能存在安全风险。
  ```
  --install-username=root  --install-usergroup=root  --install-for-all
  ```
- 如果服务器无BMC时，只能采用默认路径安装驱动。如果采用指定路径安装，ipmi_si驱动加载时会出现阻塞，导致磁盘挂载慢，影响npu正常运行。
- Device侧系统类日志通过msnpureport工具传输到Host侧进行查看。导出操作以及导出日志的存储路径具体请参见对应产品和版本的《黑匣子日志参考》的“msnpureport工具使用”章节。容器内不支持查看Device侧系统类日志，也不支持通过msnpureport工具导出Device侧系统类日志。
- 安装驱动过程中出现：“driver ko vermagic(*) is different from the os(*), need rebuild driver”，请参见**8.2.5.3 源码编译安装**解决。

若系统出现如下关键回显信息，则表示驱动安装成功。

```
Driver package installed successfully!
```

步骤7 根据系统提示信息决定是否重启系统，若需要重启，请执行以下命令；否则，请跳过此步骤。

reboot

步骤8 执行**npu-smi info**查看驱动加载是否成功。

若出现如下回显信息，说明加载成功。否则，说明加载失败。请联系华为技术支持处理。

```
+------------------------------------------------------------------------------------------+
| npu-smi 21.0.4                  Version: 21.0.4.B030                                     |
+-------------------+-----------------+----------------------------------------------------------+
| NPU     Name      | Health          | Power(W)        Temp(C)           Hugepages-Usage(page) |
| Chip    Device    | Bus-Id          | AICore(%)       Memory-Usage(MB)                        |
+===================+=================
+==============================================================+
| 6       310       | OK              | 12.8            73          0       / 970               |
| 0       3         | 0000:89:00.0    | 0               2703 / 8192                             |
+-------------------+-----------------+----------------------------------------------------------+
| 6       310       | OK              | 12.8            74          0       / 970               |
| 1       4         | 0000:8A:00.0    | 0               2867 / 8192                             |
+-------------------+-----------------+----------------------------------------------------------+
| 6       310       | OK              | 12.8            70          0       / 970               |
| 2       5         | 0000:8B:00.0    | 0               2867 / 8192                             |
+-------------------+-----------------+----------------------------------------------------------+
| 6       310       | OK              | 12.8            65          0       / 970               |
| 3       6         | 0000:8C:00.0    | 0               2867 / 8192                             |
+===================+=================
+==============================================================+
```

说明

- 加载失败，可以执行**dmesg**命令查看linux启动日志。如果出现“/*安装路径*/driver/device/davinci_mini.fd copy err” 信息，请参考**8.2.7.2 卸载驱动（适用于.run格式）**章节卸载驱动，并使用默认路径重新进行驱动安装。
- 回显中“npu-smi”后面的字段为npu-smi工具版本号；“Version:”后面的字段为驱动版本号。

----结束

8.2.6 安装固件

本章节主要介绍固件在宿主机上的安装方法，不支持在docker容器内安装。

8.2.6.1 安装固件（适用于.run 格式）

首次安装请按照“**驱动->固件**”的顺序；覆盖安装或升级请按照“**固件->驱动**”的顺序，分别安装软件包，驱动安装请参见**8.2.5 安装驱动**，本章节以*{product name}*-npu-firmware_*<version>*.run为例介绍宿主机上.run格式固件包的安装方法。

注意事项

- 在软件包安装或升级过程中，不允许对Host或Device进行复位或下电操作，否则会出现Device无法正常启动或者升级失败。
- 扩容或备件替换场景下，请执行**npu-smi info -t board -i** *id* 命令查询相应的版本号，建议服务器上所有Atlas 300I 推理卡的固件版本保持一致，若不一致，请先进行固件升级，否则可能导致服务器无法识别Atlas 300I 推理卡或Atlas 300I 推理卡在业务运行过程中出现丢卡等问题。

操作步骤

步骤1 将**8.2.3 获取软件包**准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTY登录服务器的OS命令行。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 执行如下命令，进入软件包所在路径（如“/opt”）。

cd */opt*

步骤5 执行如下命令，增加软件包的可执行权限。

chmod +x *{product name}-npu-firmware_<version>.run*

步骤6 执行如下命令，校验run安装包的一致性和完整性。

./*{product name}-npu-firmware_<version>.run* **--check**

出现如下回显信息，表示软件包校验成功。

```
Verifying archive integrity...  100%   SHA256 checksums are OK. All good.
```


版权所有 © 华为技术有限公司

说明

软件包校验使用SHA256加密方式；校验过程中若提示：*./{product name}-npu-firmware_<version>.run* does not contain an embedded MD5 checksum.和*./{product name}-npu-firmware_<version>.run* does not contain a CRC checksum，表示没有使用MD5和CRC加密方式，可忽略。

步骤7 执行如下安装命令完成安装。

./*{product name}-npu-firmware_<version>.run* **--full**

若系统出现如下关键回显信息，表示固件安装成功，并根据提示信息决定是否立即重启系统。

```
Firmware package installed successfully!
```

说明

- 若重新安装固件后安装配套版本的驱动，则无需重启系统，驱动安装后固件版本会自动生效。
- 若单独安装固件，则需执行**reboot**命令重启系统或执行**npu-smi set -t reset**命令复位标卡或芯片生效。npu-smi命令详细请参见**10.1.3 npu-smi命令介绍（适用于1.0.11-1.0.15版本）**。需保证固件与驱动版本配套，否则可能导致无法启动的兼容性问题。
- 安装固件软件包不支持指定运行用户名和用户组，共用驱动的运行用户名和用户组。
- 安装固件软件包不支持指定路径，共用驱动的安装路径。
- 软件包默认安装路径：/usr/local/Ascend
- 安装详细日志路径：/var/log/ascend_seclog/ascend_install.log。
- 安装后软件包的安装路径、安装命令以及运行用户信息记录路径：/etc/ascend_install.info
- Device侧系统类日志通过msnpureport工具传输到Host侧进行查看。导出操作以及导出日志的存储路径具体请参见对应产品和版本的《黑匣子日志参考》的“msnpureport工具使用”章节。容器内不支持查看Device侧系统类日志，也不支持通过msnpureport工具导出Device侧系统类日志。

步骤8 执行如下命令查看芯片固件版本号。若与目标版本一致，则说明安装成功。

*/usr/local/Ascend/***driver/tools/upgrade-tool --device_index -1 --component -1 --version**

说明

如果安装驱动时指定安装路径，则命令中的“*/usr/local/Ascend*”请根据实际情况替换。

----结束

8.2.6.2 安装固件（适用于.rpm 格式）

本章以*{product name}*-npu-firmware-*<version>*-*<release>*.noarch.rpm为例介绍宿主机上.rpm格式固件包的安装方法，具体请根据实际情况进行替换。

注意事项

- 当前rpm包安装不支持修改安装路径功能，请按照正常rpm流程安装，不支持rpm包的--force命令安装升级。
- Ascend 310内部部分功能，只支持HwHiAiUser用户访问和执行，用户首次安装时系统会自动创建HwHiAiUser用户。
- 安装完成后需要重启生效。卸载完成后，如需再次安装，需要先重启再安装。

- 如果已安装NPU 1.X.X系列版本的rpm包，在安装NPU 20.X.X及以上系列版本的rpm包前需要先卸载原版本再安装。
- 在软件包安装或升级过程中，不允许对Host或Device进行复位或下电操作，否则会出现Device无法正常启动或者升级失败。
- 扩容或备件替换场景下，请执行**npu-smi info -t board -i** *id* 命令查询相应的版本号，建议服务器上所有Atlas 300I 推理卡的固件版本保持一致，若不一致，请先进行固件升级，否则可能导致服务器无法识别Atlas 300I 推理卡或Atlas 300I 推理卡在业务运行过程中出现丢卡等问题。

操作步骤

须知

请按照“**驱动->固件**”的顺序，分别安装组件软件包。驱动安装请参见**8.2.5 安装驱动**。

步骤1 将**8.2.3 获取软件包**准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTY登录服务器的OS命令行。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 在软件包所在路径执行如下命令完成Host侧固件包安装。

rpm -ivh *{product name}*-npu-firmware-*<version>*-*<release>*.noarch.rpm

若系统出现如下关键回显信息，表示固件安装成功。

```
Firmware package installed successfully!
```

说明

- 若重新安装固件后安装配套版本的驱动，则无需重启系统，驱动安装后固件版本会自动生效。
- 若单独安装固件，则需执行**reboot**命令重启系统或执行**npu-smi set -t reset**命令复位标卡或芯片生效。npu-smi命令详细请参见**10.1.3 npu-smi命令介绍（适用于1.0.11-1.0.15版本）**。需保证固件与驱动版本配套，否则可能导致无法启动的兼容性问题。
- 软件包安装过程中，若无HwHiAiUser用户系统会自动创建该用户。

步骤5 重启系统。

- 若使用固件1.78.23.33.230版本软件包安装，需要执行**reboot**命令重启Host侧系统使固件版本生效。
- 若使用固件1.78.23.33.230以上版本软件包安装，根据提示信息决定是否立即重启系统。

  说明

  - 若重新安装固件后安装配套版本的驱动，则无需重启系统，驱动安装后固件版本会自动生效。
  - 若单独安装固件，则需执行**reboot**命令重启系统或执行**npu-smi set -t reset**命令复位标卡或芯片生效。npu-smi命令详细请参见**10.1.3 npu-smi命令介绍（适用于1.0.11-1.0.15版本）**。需保证固件与驱动版本配套，否则可能导致无法启动的兼容性问题。

步骤6 查看固件安装是否成功。

- 若使用固件1.78.23.33.230以上版本软件包安装，执行如下命令查看芯片固件版本号。若与目标版本一致，则说明安装成功。

  /usr/local/Ascend/driver/tools/upgrade-tool --device_index -1 --component -1 --version

- 若使用固件版本1.78.23.33.230软件包安装，执行如下命令查看固件是否安装成功。

  d-agent -v

  回显如下类似信息，说明安装成功。否则，说明安装失败，请联系华为工程师处理。

  ```
  driver: 21.0.2
  firmware: 1.78.23.33.230
  ```

  说明

  安装过程中的日志信息输出在“/var/log/ascend_seclog/ascend_install.log”文件中，用户可以执行**vim /var/log/ascend_seclog/ascend_install.log**命令打开日志。

步骤7 若使用固件1.78.23.33.230以上版本软件包安装，请跳过此步骤；若使用固件版本1.78.23.33.230软件包安装，固件安装成功后，用户需要升级芯片固件。

- 执行**d-agent -b 0000:***91:00.0*命令完成单个芯片的固件升级。
- 所有芯片上无业务进程的情况下，执行**d-agent -a**命令完成所有芯片固件升级。

须知

- Atlas 300I 推理卡（型号 3000）芯片的ID可通过**lspci | grep d100**命令查看。
- 单个芯片固件升级大概需要30s；所有芯片固件升级持续时间具体以实际为准。需等待固件升级完成后，再查询固件版本。

执行**d-agent -g 0000:***91:00.0*命令查看升级后固件版本号。回显如下类似信息，如果版本号与目标版本一致，说明升级成功。否则，说明升级失败，请联系华为技术工程师处理。nve:1.78.23.33.230 xloader:1.78.23.33.230

----结束

8.2.6.3 安装固件（适用于.deb 格式）

dpkg 安装

步骤1 将**8.2.3 获取软件包**准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTY登录服务器的OS命令行。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 执行如下命令，进入软件包所在路径（如“/opt”）。

cd */opt*

步骤5 执行如下命令，安装deb固件包（*固件包名称请根据实际情况替换*）。

dpkg -i *{product name}-npu-firmware_<version>.deb*

若系统出现如下关键回显信息，表示固件安装成功，并根据提示信息决定是否立即重启系统。

```
Firmware package installed successfully!
```

说明

- 若重新安装固件后安装配套版本的驱动，则无需重启系统，驱动安装后固件版本会自动生效。
- 若单独安装固件，则需执行**reboot**命令重启系统或执行**npu-smi set -t reset**命令复位标卡或芯片生效。npu-smi命令详细请参见**10.1.3 npu-smi命令介绍（适用于1.0.11-1.0.15版本）**。需保证固件与驱动版本配套，否则可能导致无法启动的兼容性问题。

步骤6 执行如下命令，查询固件安装信息。

dpkg -l | grep Ascend

----结束

apt-get 安装

步骤1 执行如下命令，安装本地源中的deb固件包。

- 首次安装

  apt-get install ascend310-firmware

- 重新安装

  apt-get --reinstall install ascend310-firmware

  若系统出现如下关键回显信息，表示固件安装成功，并根据提示信息决定是否立即重启系统。

  ```
  Firmware package installed successfully!
  ```

说明

- 固件对驱动有依赖关系，执行此命令会先安装好驱动包，再装固件。
- 若重新安装固件后安装配套版本的驱动，则无需重启系统，驱动安装后固件版本会自动生效。
- 若单独安装固件，则需执行**reboot**命令重启系统或执行**npu-smi set -t reset**命令复位标卡或芯片生效。npu-smi命令详细请参见**10.1.3 npu-smi命令介绍（适用于1.0.11-1.0.15版本）**。需保证固件与驱动版本配套，否则可能导致无法启动的兼容性问题。

步骤2 执行如下命令，查询固件安装信息。

apt list ascend310-firmware

----结束

8.2.7 卸载驱动

须知

请在软件包卸载前确保无相关业务进程占用昇腾驱动节点（包含容器映射场景），否则热复位将无法生效。

如果设备已经纳管到FusionDirector，卸载驱动后，FusionDirector界面上显示的驱动版本会延迟刷新，最长为1个小时。重启设备后版本会立即刷新。

版权所有 © 华为技术有限公司

8.2.7.1 确认卸载方式

表 8-5 卸载方式说明

查询命令	说明	卸载方式
dpkg -l | grep Ascend 或**apt list ascend310***	若可以查询到驱动或固件，则说明使用deb包安装。	请参见**8.2.7.4 卸载驱动（适用于.deb格式）**或**8.2.8.3 卸载固件（适用于.deb格式）**的“dpkg卸载”或“apt-get卸载”。
rpm -qa | grep npu	若可以查询到驱动或固件，则说明使用rpm包安装。	请参见**8.2.7.3 卸载驱动（适用于.rpm格式）**或**8.2.8.2 卸载固件（适用于.rpm格式）**。
• **dpkg -l | grep Ascend** • **apt list ascend310*** • **rpm -qa | grep npu**	若使用这几种命令都查询不到驱动或固件，则说明使用run包安装。	请参见**8.2.7.2 卸载驱动（适用于.run格式）**或**8.2.8.1 卸载固件（适用于.run格式）**。

8.2.7.2 卸载驱动（适用于.run 格式）

目前发布的run包在宿主机上的卸载方式相同。本章以*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run为例介绍.run格式驱动包卸载相关操作，具体请根据实际情况进行替换。

前提条件

未安装run安装包时不能使用卸载功能。

操作步骤

步骤1 使用PuTTY登录服务器的OS命令行。

步骤2 执行如下命令，切换至root用户。

su - root

步骤3 支持以下两种卸载方式，用户根据实际情况选择其中一个方式卸载即可：

- 在任意路径执行如下命令卸载软件包：

  <install-path>/**driver/script/uninstall.sh**

版权所有 © 华为技术有限公司

说明

*<install-path>*表示软件包安装路径，可以执行**cat /etc/ascend_install.info**命令查询安装路径；请根据实际情况替换。

- 在软件包所在路径执行以下命令进行卸载：

  ./*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run **--uninstall**

须知

若业务或容器占用了davinci节点，卸载过程中系统会提示“The davinci nodes are occupied by some processes, please stop processes and install or uninstall again, details in : /var/log/ascend_seclog/ascend_install.log”报错，可以选择使用**--force**参数进行强制卸载，详细请参见**8.5.1 参数说明/常用命令**。但可能导致当前正在运行的业务受到影响。

若出现如下关键回显信息，则表示驱动卸载成功。

```
Driver package uninstalled successfully!
```

步骤4 根据系统提示信息决定是否重启服务器，若需要重启系统，请执行以下命令；否则，请跳过此步骤。

reboot

须知

驱动卸载后，npu-smi工具不可用。

----结束

8.2.7.3 卸载驱动（适用于.rpm 格式）

本章主要介绍Host侧驱动卸载的相关指导。

前提条件

未安装rpm安装包时不能使用卸载功能。

操作步骤

步骤1 使用PuTTY登录服务器的OS命令行。

步骤2 执行如下命令，切换至root用户。

su - root

步骤3 执行如下命令查看rpm驱动列表。

rpm -qa | grep npu

回显中包含“npu-driver”字段表示存在驱动，如下所示：

```
npu-driver-20.2.0-1.aarch64
```


版权所有 © 华为技术有限公司

步骤4 执行如下命令卸载驱动。

rpm -e *npu-driver-20.2.0-1.aarch64*

说明

上述命令中软件名称请根据**步骤3**的查询情况进行替换。

若出现如下关键回显信息，则表示驱动卸载成功。

```
Driver package uninstalled successfully!
```

步骤5 根据系统提示信息决定是否重启服务器，若需要重启系统，请执行以下命令；否则，请跳过此步骤。

reboot

须知

驱动卸载后，npu-smi工具不可用。

----结束

8.2.7.4 卸载驱动（适用于.deb 格式）

dpkg 卸载

步骤1 执行以下命令，查询驱动安装信息。

dpkg -l | grep Ascend

步骤2 根据**步骤1**的查询结果，执行以下命令进行卸载。

dpkg -r ascend310-driver

若出现如下关键回显信息，则表示驱动卸载成功。

```
Driver package uninstalled successfully!
```

说明

- 固件对驱动有依赖关系，执行此命令需先卸载固件包，再卸载驱动。
- 如果在卸载软件包时出现重新安装其他包的情景，说明被安装的软件包曾经安装失败，需要对安装失败的软件包重新安装，确保安装成功。安装失败软件包先参考**8.4.2 deb包安装、升级或卸载失败后的处理方式**进行处理。

步骤3 根据系统提示信息决定是否重启服务器，若需要重启系统，请执行以下命令；否则，请跳过此步骤。

reboot

须知

驱动卸载后，npu-smi工具不可用。

----结束

版权所有 © 华为技术有限公司

apt-get 卸载

步骤1 执行以下命令，查询驱动安装信息。

apt list ascend310*

步骤2 根据步骤1的查询结果，执行以下命令进行卸载。

apt-get remove ascend310-driver

若出现如下关键回显信息，则表示驱动卸载成功。

```
Driver package uninstalled successfully!
```

说明

- 固件对驱动有依赖关系，执行此命令需先卸载固件包，再卸载驱动。
- 如果在卸载软件包时出现重新安装其他包的情景，说明被安装的软件包曾经安装失败，需要对安装失败的软件包重新安装，确保安装成功。安装失败软件包先参考**8.4.2 deb包安装、升级或卸载失败后的处理方式**进行处理。

步骤3 根据系统提示信息决定是否重启服务器，若需要重启系统，请执行以下命令；否则，请跳过此步骤。

reboot

须知

驱动卸载后，npu-smi工具不可用。

----结束

8.2.8 卸载固件

8.2.8.1 卸载固件（适用于.run 格式）

驱动和固件的卸载没有先后顺序要求。驱动卸载请参见**8.2.7.2 卸载驱动（适用于.run格式）**。本章以*{product name}*-npu-firmware_*<version>*.run为例介绍固件卸载相关指导，具体请根据实际情况进行替换。

前提条件

未安装run安装包时不能使用卸载功能。

操作步骤

步骤1 使用PuTTY登录服务器的OS命令行。

步骤2 执行如下命令，切换至root用户。

su - root

步骤3 支持以下两种方式，用户根据实际情况选择其中一个方式卸载即可：

- 在任意路径执行如下命令卸载软件包：
  <install-path>/**firmware/script/uninstall.sh**

说明

*<install-path>*表示软件包安装路径，可以执行**cat /etc/ascend_install.info**命令查询安装路径；请根据实际情况替换。

- 在软件包所在路径执行以下命令进行卸载：

  ./*{product name}*-npu-firmware_*<version>*.run **--uninstall**

  若出现如下关键回显信息，则表示固件卸载成功。

  ```
  Firmware package uninstalled successfully!
  ```

步骤4 根据系统提示信息决定是否重启服务器，若需要重启系统，请执行以下命令；否则，请跳过此步骤。

reboot

----结束

8.2.8.2 卸载固件（适用于.rpm 格式）

本章主要介绍Host侧固件卸载的相关指导。

前提条件

未安装rpm安装包时不能使用卸载功能。

操作步骤

步骤1 使用PuTTY登录服务器的OS命令行。

步骤2 执行如下命令，切换至root用户。

su - root

步骤3 执行如下命令查看rpm固件列表。

rpm -qa | grep npu

回显中包含“npu-firmware”字段表示存在固件，如下所示：

```
npu-firmware-21.0.2-1.noarch
```

步骤4 执行如下命令卸载固件。

rpm -e *npu-firmware-21.0.2-1.noarch*

说明

上述命令中软件名称请根据**步骤3**的查询情况进行替换。

若出现如下关键回显信息，则表示固件卸载成功。

```
Firmware package uninstalled successfully!
```

步骤5 根据系统提示信息决定是否重启服务器，若需要重启系统，请执行以下命令；否则，请跳过此步骤。

reboot

----结束

8.2.8.3 卸载固件（适用于.deb 格式）

dpkg 卸载

步骤1 执行以下命令，查询固件安装信息。

dpkg -l | grep Ascend

步骤2 根据步骤1的查询结果，执行以下命令进行卸载。

dpkg -r ascend310-firmware

若出现如下关键回显信息，则表示固件卸载成功。

```
Firmware package uninstalled successfully!
```

说明

如果在卸载软件包时出现重新安装其他包的情景，说明被安装的软件包曾经安装失败，需要对安装失败的软件包重新安装，确保安装成功。安装失败软件包先参考**8.4.2 deb包安装、升级或卸载失败后的处理方式**进行处理。

----结束

apt-get 卸载

步骤1 执行以下命令，查询固件安装信息。

apt list ascend310*

步骤2 根据步骤2的查询结果，执行以下命令进行卸载。

apt-get remove ascend310-firmware

若出现如下关键回显信息，则表示固件卸载成功。

```
Firmware package uninstalled successfully!
```

说明

如果在卸载软件包时出现重新安装其他包的情景，说明被安装的软件包曾经安装失败，需要对安装失败的软件包重新安装，确保安装成功。安装失败软件包先参考**8.4.2 deb包安装、升级或卸载失败后的处理方式**进行处理。

----结束

8.3 常用操作

8.3.1 通过 PuTTY 登录（网口方式）

操作场景

该任务指导用户使用PuTTY通过局域网远程访问服务器，对服务器实施配置、维护操作。

说明

- 您可以访问chiark网站主页下载PuTTY软件。
- 低版本的PuTTY软件可能导致登录服务器系统失败，建议使用最新版本的PuTTY软件。

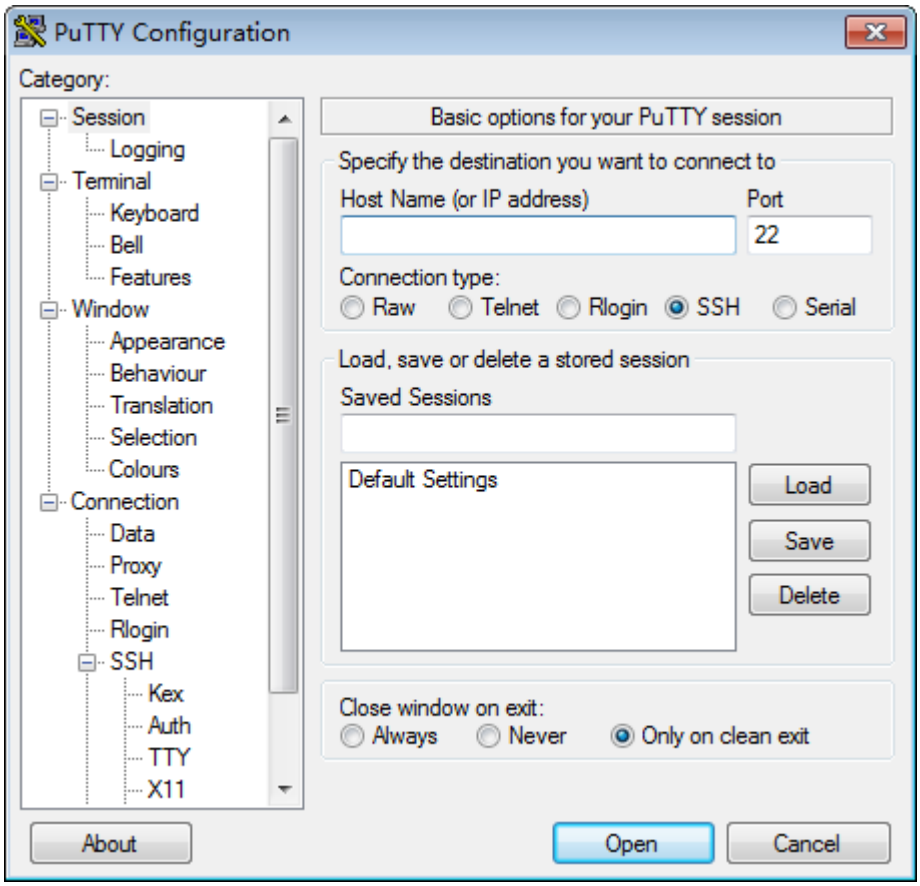
操作步骤

步骤1 设置本地PC的IP地址、子网掩码或者路由，使本地PC能和服务器网络互通。

步骤2 在本地PC上，双击“PuTTY.exe”。

弹出“PuTTY Configuration”窗口。

图 8-3 PuTTY Configuration



步骤3 在左侧导航树中，选择“Session”。

步骤4 设置登录参数。

参数说明如下：

- Host Name (or IP address)：输入要登录服务器的IP地址，如“192.168.2.10”。
- Port：默认设置为“22”。
- Connection type：默认选择“SSH”。
- Close window on exit：默认选择为“Only on clean exit”。

版权所有 © 华为技术有限公司

说明

配置“Host Name”后，再配置“Saved Sessions”并单击“Save”保存，则后续使用时直接双击“Saved Sessions”下保存的记录即可登录服务器。

步骤5 单击“Open”。

进入“PuTTY”运行界面，提示“login as:”，等待用户输入用户名。

说明

- 如果首次登录该目标服务器，则会弹出“PuTTY Security Alert”窗口。单击“是”表示信任此站点，进入“PuTTY”运行界面。
- 如果帐号输入错误，必须重新连接PuTTY。

步骤6 按提示分别输入用户名和密码。

登录完成后，命令提示符左侧显示出当前登录服务器的主机名。

----结束

8.3.2 使用 WinSCP 传输文件

操作场景

在PC机上使用WinSCP工具进行文件传输。

必备事项

前提条件

目的设备已开启SFTP服务。

数据

需准备如下数据：

- 待连接服务器的IP地址
- 登录待连接服务器的用户名和密码

软件

WinSCP.exe：此工具为第三方免费软件。

操作步骤

步骤1 打开“WinSCP”文件夹，双击“WinSCP.exe”。

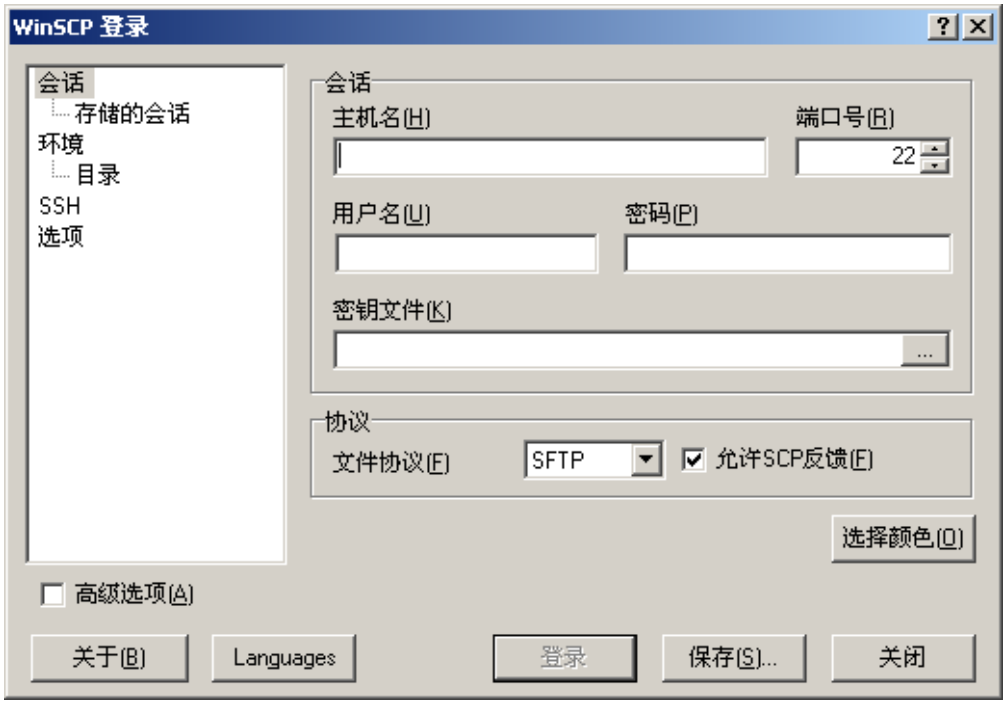
弹出“WinSCP 登录”对话框，如图8-4所示。

说明

若系统非中文操作系统，可以单击“Languages”进行界面语言的选择。

版权所有 © 华为技术有限公司

图 8-4 WinSCP 登录



步骤2 设置登录参数。

参数说明如下。

- 主机名（H）：输入待连接设备的IP地址。例如：“192.168.2.10”。
- 端口号（R）：默认为“22”。
- 用户名（U）：输入用户名。例如：admin123。
- 密码（P）：输入密码。例如：admin123。
- 密钥文件（K）：默认为空，保留默认值。
- 协议：选择默认文件协议“SFTP”，并勾选“允许SCP反馈（F）”。

步骤3 单击“登录”。

进入“WinSCP”文件传输界面。

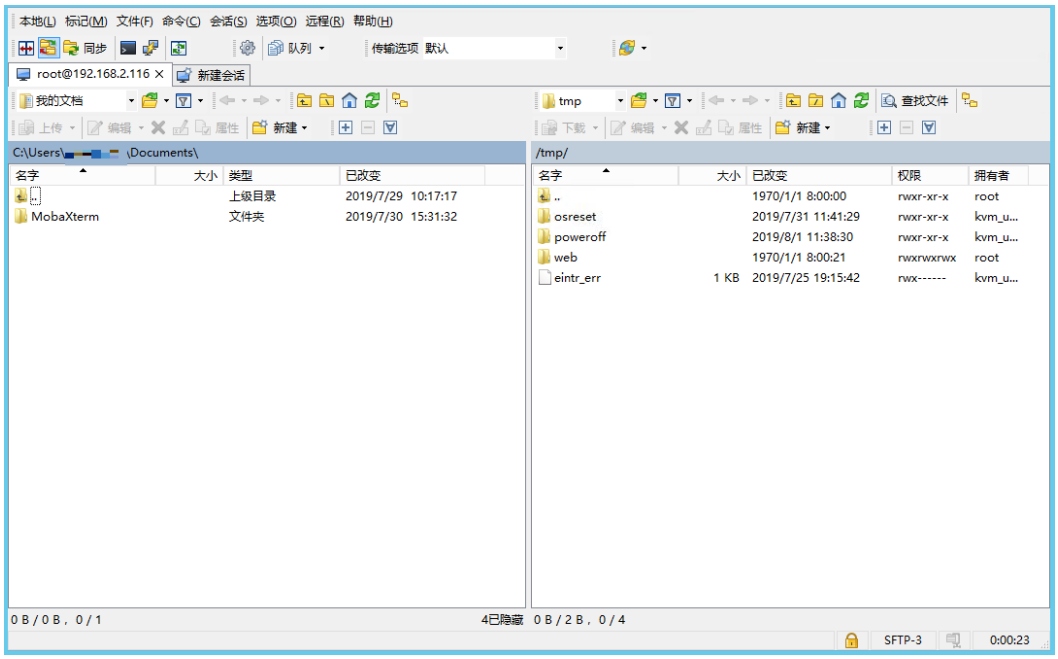
说明

- 如果首次登录时没有选择密钥文件，此时会弹出一个警告提示框，询问“是否连接并添加密钥到缓存？”，单击“是（Y）”，进入“WinSCP”文件传输界面。
- 以Windows 7为例，界面左侧区域默认打开本地计算机的“C:\Users\Administrator\Documents”目录，右侧区域默认打开设备的“/root”目录。

步骤4 根据实际需求，在界面左右区的指定目录中进行文件夹的创建、删除和复制等操作。

版权所有 © 华为技术有限公司

图 8-5 WinSCP 界面



----结束

8.3.3 设置用户有效期

为保证用户的安全性，应设置用户的有效期，使用系统命令**chage**来设置用户的有效期。

命令为：

chage [-m *mindays*] **[-M** *maxdays]* **[-d** *lastday*] **[-I** *inactive*] **[-E** *expiredate*] **[-W** *warndays*] *user*

相关参数请参见**表8-6**。

表 8-6 设置用户有效期

参数	参数说明
-m	密码可更改的最小天数。设置为“0”表示任何时候都可以更改密码。
-M	密码保持有效的最大天数。设置为“-1”表示可删除这项密码的检测，用户密码则不会超期，存在安全风险，请谨慎使用。
-d	上一次更改密码的日期。
-I	停滞时期。过期指定天数后，设定密码为失效状态。
-E	用户到期的日期。超过该日期，此用户将不可用。
-W	密码到期前，提前收到警告信息的天数。
-l	列出用户以及密码的有效期。由非特权用户来确定用户或密码何时过期。

版权所有 © 华为技术有限公司

说明

- 表8-6只列举出常用的参数，用户可通过**chage --help**命令查询详细的参数说明。
- 日期格式为YYYY-MM-DD，如**chage -E** 2019-12-01 *test*表示用户*test*的用户在2019年12月1日过期。
- User如果不填写，默认为root用户。

例如：修改用户*test*的有效期为2019年12月31日。可执行如下命令：

chage -E *2019-12-31 test*

8.3.4 配置 Host 侧 IP 地址通过 SSH 方式登录 Device

操作步骤

步骤1 使用**npu-smi set -t ssh-enable**命令打开SSH开关，命令的具体使用方法请参见**10.1.3.5.7 配置所有芯片的ssh使能状态**和**10.1.3.5.8 配置指定芯片的ssh使能状态**。

步骤2 在Host侧配置IP。

IP必须与Device侧的IP为同一网段，如：192.168.1.100。Device默认IP地址从192.168.1.199依次递减，例如用户有多个Device，则IP地址分别为192.168.1.199，192.168.1.198，192.168.1.197，...，最多支持配置64个Device。

本文仅给出Host操作系统为Ubuntu 18.04和CentOS 7.6配置IP的方法，其他OS配置IP的方法可参考对应OS官网社区的FAQ去解决。

- 在Host操作系统为Ubuntu 18.04配置动态IP（动态添加IP，每次重启后都要重新操作）
  a. 执行以下命令，配置动态IP。

     ifconfig endvnic *192.168.1.100*

  b. 执行**ifconfig**命令，查看网络配置是否成功。

     若出现如下回显，表示网络配置成功。否则，请联系华为技术支持处理。

```
endvnic: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.100  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::xxxx:xxxx:xxxx:74d3  prefixlen 64  scopeid 0x20<link>
        ...
```

- 在Host操作系统为Ubuntu 18.04配置静态IP
  a. 修改“/etc/netplan”目录下的yaml配置文件。
     i. 执行以下命令，进入“/etc/netplan”目录下查看配置文件名称。

     cd /etc/netplan

     ls

     查询结果如下：

```
01-netcfg.yaml
```

     ii. 执行以下命令，打开配置文件。

     vi /etc/netplan/*01-netcfg.yaml*

     在该文件中增加以下内容：

```
endvnic:
                addresses:   [192.168.1.100/24]
                nameservers:
                        addresses:    [255.255.254.0]
```


版权所有 © 华为技术有限公司

按“Esc”切换到命令模式，输入**:wq**保存并退出。

b. 执行以下命令，使网络配置生效。

netplan apply

c. 执行**ifconfig**命令，查看网络配置是否成功。

若出现如下回显，表示网络配置成功。否则，请联系华为技术支持处理。

```
endvnic: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.100  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::xxxx:xxxx:xxxx:74d3  prefixlen 64  scopeid 0x20<link>
        ...
```

- 在Host操作系统为CentOS 7.6配置动态IP（动态添加IP，每次重启后都要重新操作）

  a. **systemctl status NetworkManager**

  如果不是active的，需要启动**systemctl start NetworkManager**

  b. **ifconfig -s endvnic 192.168.1.100**

  c. **systemctl disable NetworkManager**

  d. **systemctl stop NetworkManager**

- 在Host操作系统为CentOS 7.6配置静态IP（静态设置，每次重启无需重复操作）

  a. **systemctl status NetworkManager**

  如果不是active的，需要通过如下命令配置开机自启动。

  systemctl start NetworkManager

  systemctl enable NetworkManager

  b. **nmcli connect add autoconnect no type ethernet con-name endvnic ifname endvnic** 生成“/etc/sysconfig/network-scripts/ifcfg-endvnic”。

  c. 使用**ip a** 查看MAC地址，例如：

```
3: endvnic: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
1000
link/ether 10:1b:54:xx:xx:d3 brd ff:ff:ff:ff:ff:ff
inet 192.168.1.100/24 brd 192.168.1.255 scope global endvnic
valid_lft forever preferred_lft forever
inet6 fe80::xxxx:xxxx:xxxx:f0d3/64 scope linkifconfig
valid_lft forever preferred_lft forever
其中 10:1b:54:xx:xx:d3 为MAC地址
```

须知

虚拟网口MAC地址说明

Driver包安装后生成的虚拟网口没有固定的MAC地址，Driver包升级或安装后生成随机数作为MAC地址，虚拟网口MAC地址固定格式：10:1b:54:xx:xx:d3，其中第四和第五位为随机数，10:1b:54前三位MAC地址属于华为公司。

Driver包升级或安装后，虚拟网口MAC地址会变化，需要重新进行MAC地址与IP地址的绑定。

  d. 修改“/etc/sysconfig/network-scripts/ifcfg-endvnic”内容（请确保该路径下只有ifcfg-endvnic一个文件，备份文件或重命名文件例如ifcfg-endvnic1，都会影响添加网卡的结果）。

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
```


版权所有 © 华为技术有限公司

```
#BOOTPROTO=dhcp  #注释掉
DEFROUTE=yes
#IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=endvnic
UUID=6c2db480-1993-40ba-bc21-3541ae728719
DEVICE=endvnic
IPADDR=192.168.1.100  #配置为Host侧虚拟网卡地址
ONBOOT=yes
HWADDR="10:1b:54:xx:xx:d3"  #配置虚拟网卡MAC地址
其中，HWADDR为MAC地址
```

e. 使用**ifup endvnic**命令添加网卡。

f. 执行如下命令，查看网络配置是否成功。

ifconfig

若出现如下回显，表示网络配置成功。否则，请联系华为技术支持处理。

```
endvnic: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.100  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::xxxx:xxxx:xxxx:78d3  prefixlen 64  scopeid 0x20<link>
        ether 10:1b:54:xx:xx:d3  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 42  bytes 2940 (2.9 KB)
        TX errors 8  dropped 8 overruns 0  carrier 0  collisions 0
```

步骤3 登录Device。通过**ssh HwHiAiUser@*xxx*** 命令登录device（xxx为device侧IP地址，如：192.168.1.199）。

说明

- Device重启后，Device侧的配置文件会恢复默认配置（默认账号的超期时间恢复为90天），若用户需要自定义过期时间，可以修改配置文件“/etc/login.defs”中有效期的天数；或者建议用户在自定义的启动脚本中添加对应的chage命令，重新配置过期时间，详情请参见**8.3.3 设置用户有效期**。
- 首次登录Device侧需要修改密码，默认用户名和密码具体请参见**《Atlas硬件产品 用户清单》**。新密码设置规则：密码必须包含字母、数字、特殊字符（特殊字符不能用**@**、**%**、**空格**），长度不小于8个字符。

----结束

8.4 FAQ

8.4.1 升级系统内核版本时未卸载 run 包导致 run 包不可用

问题描述

如果在安装了run包的环境直接升级内核，再次重启时环境会出现启动异常，如**图8-6**所示。如果后续在该环境上再安装run包也无法使用。这是由于安装run包时驱动ko加载到内核，而升级内核版本会触发根文件系统打包，会将原来环境中ko打包到根文件系统，导致启动异常。

图 8-6 run 包不可用

```
[    3.690421] i8042: No controller found
[    5.850142] systemd[1]: Failed to start Load Kernel Modules.
[FAILED] Failed to start Load Kernel Modules.
See 'systemctl status systemd-modules-load.service' for details.
[  OK  ] Started Create Static Device Nodes in /dev.
         Starting Apply Kernel Variables...
[  OK  ] Started Apply Kernel Variables.
[  OK  ] Started Setup Virtual Console.
[  OK  ] Started Journal Service.
[  OK  ] Started dracut cmdline hook.
         Starting dracut pre-udev hook...
[  OK  ] Started dracut pre-udev hook.
         Starting udev Kernel Device Manager...
[  OK  ] Started udev Kernel Device Manager.
         Starting udev Coldplug all Devices...
         Mounting Configuration File System...
[  OK  ] Mounted Configuration File System.
[    6.861746] megaraid_sas 0000:18:00.0: IOC INIT command return status SUCCESS for SCSI host 0
[  OK  ] Started udev Coldplug all Devices.
         Starting Show Plymouth Boot Screen...
[  OK  ] Reached target System Initialization.
         Starting dracut initqueue hook...
[    7.059120] megaraid_sas 0000:18:00.0: return from megasas_get_target_prop 7453 return value 2
[    8.777736]  [drv_pcie] [devdrv_get_boot_mode_flag 390] get boot mode from bios err:0x0
[    8.777978]  [drv_pcie] [devdrv_load_file_copy 457] devdrv_device_driver, unable to open file: /home/bios/davinci_cloud.image
 (-5)
[    8.778341]  [drv_pcie] [devdrv_load_file_trans 632] devdrv_device_driver, /home/bios/davinci_cloud.image file copy err
[    8.778909]  [drv_pcie] [devdrv_load_file 1045] devdrv_device_driver, trans file to agent bios failed
[    8.811736]  [drv_pcie] [devdrv_get_boot_mode_flag 390] get boot mode from bios err:0x0
[    8.811976]  [drv_pcie] [devdrv_load_file_copy 457] devdrv_device_driver, unable to open file: /home/bios/davinci_cloud.image
 (-5)
[    8.812341]  [drv_pcie] [devdrv_load_file_trans 632] devdrv_device_driver, /home/bios/davinci_cloud.image file copy err
[    8.812901]  [drv_pcie] [devdrv_load_file 1045] devdrv_device_driver, trans file to agent bios failed
```

解决方案

步骤1 手动卸载ko。

1. 查询当前系统存在ko列表。

   lsmod|grep drv

2. 将查询到的ko一次全部卸载，多个ko名字之间需要加空格。两个ko的示例如下（ko1，ko2请用实际ko名称替换）。

   rmmod *ko1 ko2*

步骤2 打包根文件系统。

dracut --force

步骤3 重启环境。

reboot

----结束

8.4.2 deb 包安装、升级或卸载失败后的处理方式

问题描述

deb包安装、升级或卸载失败后，残留的缓存信息会导致再次安装或升级异常。

解决方案

deb包安装、升级或卸载失败后，需要手动删除缓存信息等相关操作，步骤如下：

步骤1 删除失败包的配置文件列表，执行命令如下（以ascend310-driver包为例）。

```
rm /var/lib/dpkg/info/ascend310-driver*
```

步骤2 执行强制删除命令。

```
dpkg --remove --force-remove-reinstreq ascend310-driver
```


版权所有 © 华为技术有限公司

步骤3 删除原安装失败软件目录下的残留文件（示例目录为默认安装路径，以实际安装目录为准）。

```
rm -rf /usr/local/Ascend/driver
```

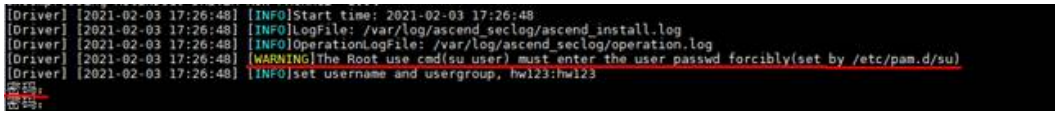
----结束

8.4.3 root 用户切换到普通用户需要输入密码权限

问题描述

在软件包安装过程中，可能会出现“The Root use cmd(su user) must enter the user passwd forcibly(set by /etc/pam.d/su)”报错。如图8-7所示：

图 8-7 报错信息



可能原因

root用户执行**su**命令进行用户切换时需要输入用户密码。

解决方案

- 直接输入运行用户密码继续完成后续安装。
- 修改系统配置文件“/etc/pam.d/su”，设置root用户执行**su**命令进行用户切换时不需要输入用户密码。操作如下：
  a. 执行如下命令，打开配置文件“/etc/pam.d/su”。

     vi /etc/pam.d/su

  b. 将“auth sufficient pam_rootok.so”前面"**#**"号删除，保存退出。

8.4.4 如何检查 device 是否正常运行

步骤1 登录运行环境，并在root用户下执行以下操作。

步骤2 查询run包安装路径。

cat /etc/ascend_install.info

系统显示示例如下：

```
Driver_Install_Path_Param=/usr/local/Ascend
```

步骤3 进入Driver安装路径，通过upgrade-tool工具查看下device侧运行文件系统版本。

cd */usr/local/Ascend/*driver/tools/

./upgrade-tool --device_index -1 --system_version

如果能正常查询，则说明device侧已经正常启动，正常查询返回信息类似如下。

```
[root@localhost tools]# /usr/local/Ascend/driver/tools/upgrade-tool --device_index -1 --system_version
{
Get system version[ ] succeed, deviceId(0)
        {"device_id":0, "version"[ ]
Get system version[ ]) succeed, deviceId(1)
        {"device_id":1, "version":[ ]
}
```

----结束

8.4.5 复位或下电导致 Device 无法正常启动

问题描述 1

在软件包安装或升级过程中，对Device进行了复位或下电操作，出现固件损坏，导致Device无法正常启动。

解决方案 1

1. 对Device连续进行3次复位（非下电，使用**reboot**命令）操作，3次复位之间间隔建议10秒以上。
2. 如Device仍无法启动，则对Device进行下电重启。

问题描述 2

在软件包安装或升级过程中，对Host进行了复位或下电操作，导致升级失败或Device无法正常启动。

解决方案 2

1. 若Device能正常启动，重新升级软件包。
2. 若Device无法正常启动，解决方法请参见**解决方案1**。

8.4.6 软件包格式混用安装或升级恢复方法

问题描述

当前环境已安装run包，覆盖安装或升级时使用rpm包或者deb包。或当前环境已安装rpm包或者deb包，覆盖安装或升级时使用run包。

解决方案

环境已安装软件包格式	覆盖安装或升级的软件包格式	处理方法
.rpm	.run	1. 卸载rpm包，具体请参见**8.2.7.3 卸载驱动（适用于.rpm格式）**和**8.2.8.2 卸载固件（适用于.rpm格式）**。 2. 重新安装软件包，具体请参见**8.2.5 安装驱动**和**8.2.6 安装固件**。
.run	.rpm	

版权所有 © 华为技术有限公司

环境已安装软件包格式	覆盖安装或升级的软件包格式	处理方法
.deb	.run	1. 卸载deb包，具体请参见**8.2.7.4 卸载驱动（适用于.deb格式）**和**8.2.8.3 卸载固件（适用于.deb格式）**。 若卸载失败，请参见**8.4.2 deb包安装、升级或卸载失败后的处理方式**处理。 2. 重新安装软件包，具体请参见**8.2.5 安装驱动**和**8.2.6 安装固件**。
.run	.deb	

8.4.7 安装软件包时提示解压失败

问题描述

软件包安装时出现“Extraction failed”或“Decompression failed”报错。如**图8-8**所示：

图 8-8 报错打印

```
Verifying archive integrity...  100%   SHA256 checksums are OK. All good.
Uncompressing npu-driver-run-package ... Extraction failed.
 ... Decompression failed.
  100%  Signal caught, cleaning up
```

可能原因

系统缺少tar工具包。

执行**tar**命令，若出现如下回显，则说明缺少tar工具包。

```
-bash: tar: command not found
```

解决方案

请根据相应操作系统自行安装tar工具。

8.4.8 驱动加载超时，CPU 打印“soft lockup”报错

问题描述

加载驱动出现概率性system-udevd服务加载超时和“soft lockup”报错。

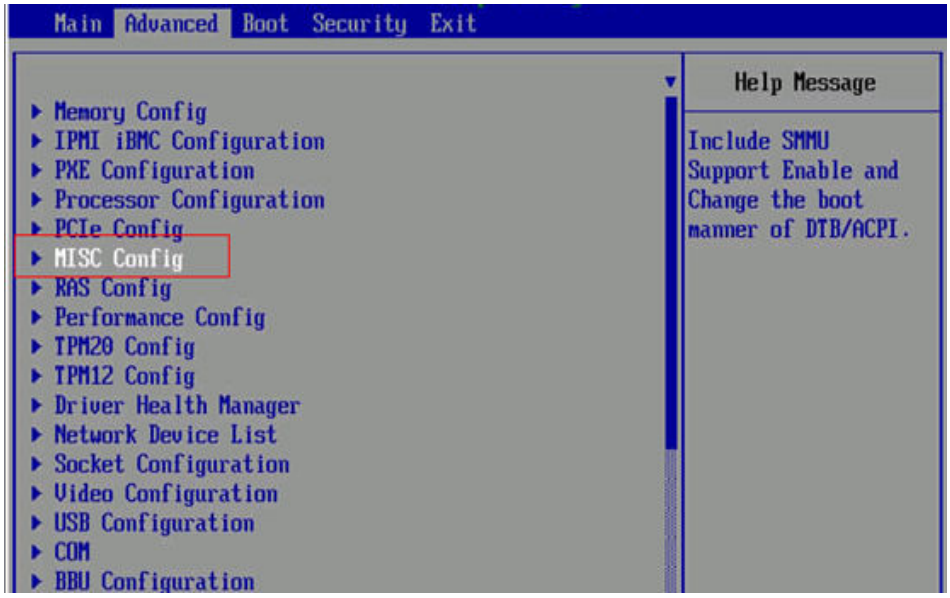
可能原因

在开启BMC串口日志重定向后，当前系统默认将内核printk日志级别设置为DEBUG，因此驱动加载过程中的日志会打印到串口，导致CPU调度变慢，出现概率性system-udevd服务加载超时和“soft lockup”报错。

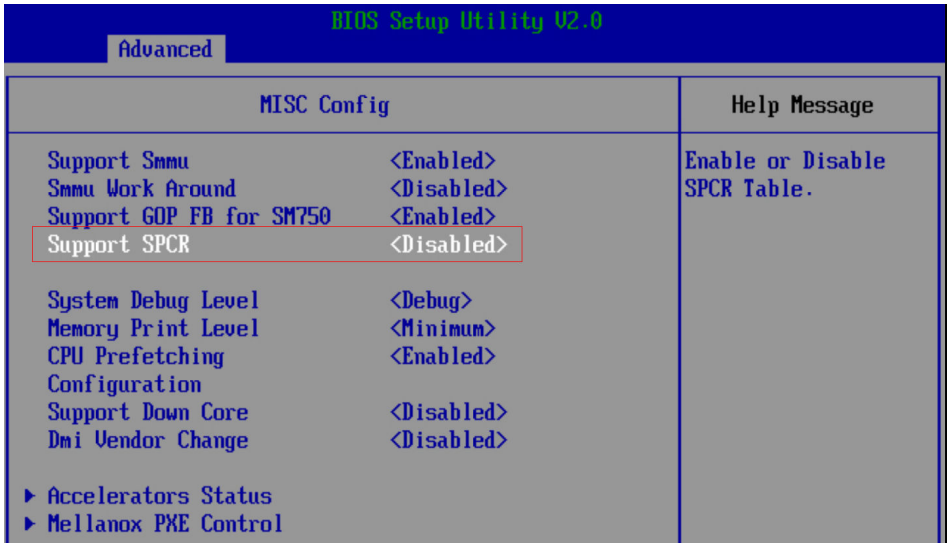
版权所有 © 华为技术有限公司

解决方案

- Atlas 800 推理服务器（型号：3000）环境关闭BMC串口日志重定向具体操作如下：
  a. 系统启动时按Delete键进入BIOS设置界面。
  b. 选择“Advanced > MISC Config”。



  c. 将“Support SPCR”设置为“Disabled”。



  d. 按“F10”。
  e. 弹出“Save configuration changes and exit? ”对话框。
  f. 选择“Yes”，按Enter键保存BIOS参数设置，此时服务器会自动重启。

版权所有 © 华为技术有限公司

8.4.9 驱动卸载过程中出现进程占用报错

问题描述

驱动卸载过程中出现“The davinci nodes are occupied by some processes, please stop processes and install or uninstall again, details in : /var/log/ascend_seclog/ascend_install.log”报错，如图8-9所示。

图 8-9 报错信息

```
Verifying archive integrity...  100%   SHA256 checksums are OK. All good.
Uncompressing ASCEND910 DRIVER RUN PACKAGE  100%
[Driver] [2022-03-24 22:28:23] [INFO]Start time: 2022-03-24 22:28:23
[Driver] [2022-03-24 22:28:23] [INFO]LogFile: /var/log/ascend_seclog/ascend_install.log
[Driver] [2022-03-24 22:28:23] [INFO]OperationLogFile: /var/log/ascend_seclog/operation.log
[Driver] [2022-03-24 22:28:23] [INFO]base version is 1.81.22.1.234.
[Driver] [2022-03-24 22:28:23] [WARNING]Do not power off or restart the system during the installation/upgrade
[Driver] [2022-03-24 22:28:23] [INFO]set username and usergroup, HwHiAiUser:HwHiAiUser
[Driver] [2022-03-24 22:28:24] [INFO]driver and firmware version relationship check success
[Driver] [2022-03-24 22:28:24] [INFO]Driver package has been installed on the path /usr/local/Ascend, the version is 1.81.22.1.234, and the version of this package is 1.81.22.1.2
34,do you want to continue?  [y/n]
y
[Driver] [2022-03-24 22:28:29] [ERROR]The davinci nodes are occupied by some processes, please stop processes and install or uninstall again, details in : /var/log/ascend_seclog/
ascend_install.log
[Driver] [2022-03-24 22:28:29] [INFO]If you want to install the driver forcibly, add the force parameter. For details, see [--help].
[Driver] [2022-03-24 22:28:29] [INFO]End time: 2022-03-24 22:28:29
```

可能原因

用户业务进程未关闭，卸载过程被拦截。

解决方案

说明

若卸载rpm、deb包时，提示“The davinci nodes are occupied by some processes”，请停止业务进程后重新卸载驱动。

步骤1 执行如下命令查看“/var/log/ascend_seclog/ascend_install.log”日志，确定未关闭的业务进程及其占用的节点信息。

cat /var/log/ascend_seclog/ascend_install.log

回显信息如下：

```
[Driver] [2022-03-25 01:10:49] [WARNING]The setting hot reset flag does not take effect.
[Driver] [2022-03-25 01:10:49] [WARNING]/dev/davinci_manager has user process:  47355
 root      47355  23097  0 01:10 pts/1    00:00:00 ./hlt_hdc_test 65 0 6 100
[Driver] [2022-03-25 01:10:51] [ERROR]The davinci nodes are occupied by some processes, please stop
processes and install or uninstall again.
```

步骤2 执行以下操作继续卸载驱动。

- 停止业务进程后，重新卸载驱动。具体请参见8.2.7 卸载驱动。
- 不停止业务使用**--force**参数进行强制卸载驱动，具体操作如下。但正在运行的业务可能会受到影响，且执行后必须重启host驱动才可以生效。
  - 使用软件包卸载。

    ./*{product name}*-npu-driver_*x.x.x*_linux-*{arch}*.run **--uninstall --force**
  - 在任意路径执行如下命令卸载。

    *<install-path>***/driver/script/uninstall.sh --force**

    *<install-path>*表示软件包安装路径，可以执行**cat /etc/ascend_install.info**命令查询安装路径；请根据实际情况替换。

----结束

版权所有 © 华为技术有限公司

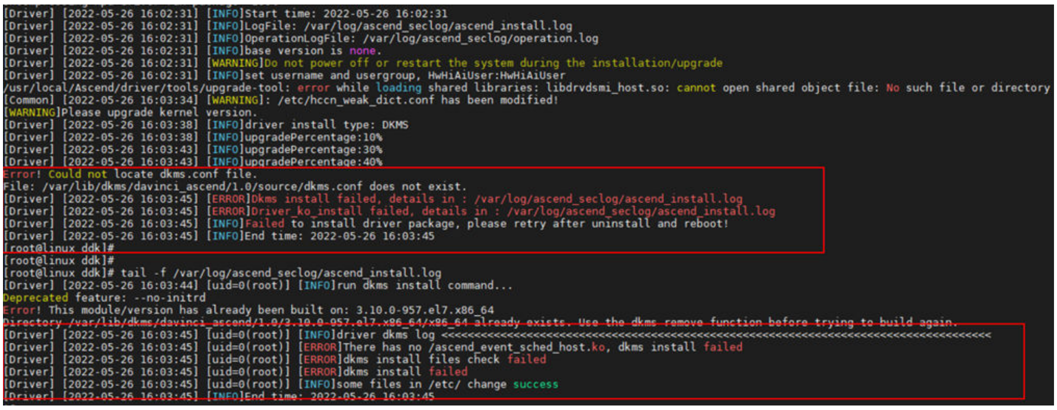
8.4.10 驱动安装过程中出现 dkms 编译失败报错

问题描述

驱动安装过程中出现类似如下报错。

```
[ERROR]Dkms install failed, details in : var/log/ascend_seclog/ascend_install.log."
"[ERROR]Driver_ko_install failed, details in : /var/log/ascend_seclog/ascend_install.log.
```

图 8-10 报错信息



可能原因

由于dkms工具问题导致驱动编译失败后残留。

解决方案

步骤1 进入"/var/lib/dkms"目录。

步骤2 删除"davinci_ascend"目录。

步骤3 重新安装驱动。

----结束

8.4.11 安装 deb 包时出现"binary opetator expected"报错

问题描述

安装deb包失败，出现"binary operator expected"报错，如**图8-11**所示。

图 8-11 安装失败报错

```
install firmware run fail, now to uninstall run pkg
Usage: ./[redacted]-npu-firmware_1.81.22.5.220.run [options]
Options:
  --help | -h                         Print this message
  --info                              Print embedded info : title, default target directory, embedded script ...
  --list                              Print the list of files in the archive
  --check                             Checks integrity and version dependency of the archive
  --quiet                             Quiet install mode, skip human-computer interactions
  --nox11                             Do not spawn an xterm
  --noexec                            Do not run embedded script
  --extract=<path>                    Extract directly to a target directory (absolute or relative)
                                      Usually used with --noexec to just extract files without running
  --tar arg1 [arg2 ...]               Access the contents of the archive through the tar command
  --full                              Install full mode
  --run                               Install run mode
  --upgrade                           Upgrade product immediately
  --reset                             Resetting user config
  --uninstall                         Uninstall product
  --version                           Query the package version

/var/lib/dpkg/info/[redacted]-firmware.postinst: line 3: [: ./[redacted]-npu-firmware_1.81.22.5.220.run: binary operator expected
dpkg: error processing package ascend310-firmware (--install):
 installed [redacted]-firmware package post-installation script subprocess returned error exit status 1
Errors were encountered while processing:
 [redacted]-firmware
```

可能原因

安装deb包时，会将deb包内的run包解压到当前运行环境的根目录“./”下，若用户根目录下已存在相同版本的run包，则会使安装脚本检索到2个安装包，出现“binary operator expected”报错。

解决方案

移除根目录“./”下的run包，重新安装软件包。

8.5 参考

8.5.1 参数说明/常用命令

参数说明

软件包支持根据命令行完成一键式安装，各个命令之间可以配合使用，用户根据安装需要选择对应参数完成安装，所有参数都是可选参数。

安装命令格式： **./***xx*.run [options]

详细参数请参见**表8-7**。

须知

软件包中的**xx**请根据实际情况进行替换。

如果通过**./***xx*.run **--help**命令查询出的参数未解释在如下表格，则说明该参数预留或适用于其他芯片版本，用户无需关注。

表 8-7 安装包支持的参数说明

参数	说明
--run	运行态类型：只安装运行场景需安装的文件。

版权所有 © 华为技术有限公司

参数	说明
--devel	开发态类型：包含用户开发时需要用到的头文件。 Firmware子包不支持该参数。
--full	完整安装类型：安装所有文件。
--docker	容器场景下安装，仅用于driver子包，其他子包默认按full模式。
--install-username=<username>	首次安装：可以指定运行用户名，若不指定，则默认是HwHiAiUser。 覆盖安装：沿用上次运行用户名。 **说明** ● 该参数需要配合“--install-usergroup=<usergroup>”一起使用，并且username需要与创建的用户（**8.2.4 创建运行用户**）保持一致。 ● 如果指定root用户为运行用户，则需要与--install-for-all参数配合使用，如下所示，该场景下权限控制可能存在安全风险。 `--install-username=root --install-usergroup=root --install-for-all` ● Firmware安装不支持指定运行用户名和用户组，共用Driver的运行用户名和用户组。
--install-usergroup=<usergroup>	首次安装：可以指定运行用户组，若不指定，则默认是HwHiAiUser。 覆盖安装：沿用上次运行用户组。 **说明** 该参数需要配合“--install-username=<username>”一起使用，并且usergroup需要与创建的用户组（**8.2.4 创建运行用户**）保持一致。
--install-path=<path>	指定安装目录，如不指定： ● 若使用root用户安装，默认安装路径为：/usr/local/Ascend。 ● 若使用非root用户安装，则默认安装路径为：$HOME/Ascend。 若通过该参数指定了安装路径，运行用户需要对指定的安装路径有可读写权限。安装路径支持大小写字母（a-z，A-Z）、数字（0-9）、下划线（_）、中划线（-）、句点（.（非相对路径））、单个/（文件名或目录不支持/）。 **说明** Firmware安装不支持指定路径，共用Driver的安装路径。

版权所有 © 华为技术有限公司

参数	说明

--install-for-all	安装时，允许其他用户具有安装群组的权限。
	当安装携带该参数时，软件包中创建的目录及文件，其他用户权限=安装群组权限。
	首次安装过程中如果使用了--install for all参数，覆盖安装中也需要保留--install for all参数。
	使用示例：**./***xx*.run **--full --install-for-all**
	说明
	● 使用该参数将会存在安全风险：其他所有用户都有权限访问安装目录，请谨慎使用。
	● Firmware不支持该参数。
	● 安装时如果使用该参数：
	  ● 如果安装路径不存在，则安装过程中会自动创建默认路径，例如$HOME/Ascend，该路径权限为755，其他用户可以正常使用。
	  ● 如果安装路径已存在，则请确保该目录权限至少为755，否则将会终止安装。

--reset	恢复初始化配置（该参数只适用于Firmware子包）： ● --full --reset参数通过安装方式恢复初始化配置，需立即重启生效；若安装失败，则恢复初始化配置失败。 ● --upgrade --reset参数通过升级方式恢复初始化配置，需立即重启生效；若升级失败，则恢复初始化配置失败。 恢复初始化配置后，原来用户对device侧的个人配置数据会失效，个人配置数据仅包含ssh登录密码和ssh使能状态，需要用户重新配置。 恢复初始化配置后，请务必立即修改Device的ssh登录密码（包括HwHiAiUser用户和root用户的密码）。 **说明** ● --reset参数只支持--full --reset与--upgrade --reset两种模式的使用方法，不支持与其它参数配合使用，也不支持单独使用。 ● 带有--reset参数命令执行完成后，重启生效前，禁止通过ssh、scp等方式访问Device侧的动作，否则会造成恢复初始化配置失败。
--uninstall	卸载。
--nox11	安装过程中不弹出图形终端窗口。 使用示例：**./***xx*.run **--full --nox11** **说明** 该参数仅支持NPU 21.0.4及以上版本。
--noexec	不执行安装脚本。配套--extract=*path*使用。格式为：--noexec --extract=*path*
--extract=*<path>*	解压缩安装包中文件到指定目录。使用该参数时，用户无需手动建立指定的目录，解压过程中会自动创建，例如： --noexec --extract=./package，则命令执行后会自动将解压后的内容放在package目录。

版权所有 © 华为技术有限公司

参数	说明
--upgrade	升级。 ● 首次安装场景，系统默认安装到“/usr/local/Ascend”目录。 ● 非首次安装场景，仅支持从软件包所在路径升级。 **说明** 该参数不能与--install-username、--install-usergroup、--install-for-all其中一个参数配合使用。 默认支持容器算力分组场景。
--help或者-h	查看帮助。
--check	校验软件包的完整性、检查与已经安装的其他软件包之间的配套关系是否正确。
--version	查询版本号。
--tar arg1 [arg2 …]	对安装包执行tar命令，使用tar后面的参数作为命令的参数。例如执行**--tar xvf**命令，解压软件包的内容到当前目录。
--list	导出安装包文件列表。
--info	显示数据包详细信息。
--quiet	静默安装，跳过交互式信息。 使用该参数进行安装时，若安装目录权限大于755，则会提示安全风险，用户自行选择终止安装或选择其他权限为755的安装目录重新安装。 使用示例：**./***xx*.run **--full --quiet** **说明** Driver和Firmware软件包在使用该参数时，不能与--uninstall配合使用。
--repack [package_name]	构建新驱动包。 package_name为新构建的驱动包名称，格式为“*xx*.run”。 **说明** 若package_name参数为空，则会在当前路径下生成一个命名为“原驱动包名称-custom.run”的文件。
--repack-path=<path> [package_name]	在指定路径下构建新驱动包。 ● path为原run包的解压目录。 ● package_name为新构建的驱动包名称，格式为“*xx*.run”。 **说明** 若package_name参数为空，则会在当前路径下生成一个命名为“原驱动包名称-custom.run”的文件。

版权所有 © 华为技术有限公司

参数	说明
--force	控制仅安装不直接生效。使用该参数安装软件包时需要重启生效。（该参数仅适用于driver子包） 该参数需要与--run、--full、--upgrade、--uninstall等其中一个参数配合使用，例如./*xx*.run **--full --force** **说明** 该参数仅支持NPU 22.0.0及以上版本。 若首次安装驱动使用该参数时，必须重启后才能卸载或覆盖安装，否则可能会导致卸载或覆盖安装失败。

举例说明：

- full模式下安装
  - 不指定安装目录：**./***xx*.run **--full**
  - 指定安装目录：**./***xx*.run **--full --install-path=***安装路径*
- docker模式下安装（仅支持driver子包）
  - 不指定安装目录：**./***xx*.run **--docker**
  - 指定安装目录**./***xx*.run **--docker --install-path=***安装路径*
- run模式下安装
  - 不指定安装目录：**./***xx*.run **--run**
  - 指定安装目录：**./***xx*.run **--run --install-path=***安装路径*
- 开发态模式下安装
  - 不指定安装目录：**./***xx*.run **--devel**
  - 指定安装目录：**./***xx*.run **--devel --install-path=***安装路径*

8.5.2 驱动安装过程中所需的 Linux 工具

驱动安装过程中所需的Linux工具见表8-8。

表 8-8 驱动安装过程中所需的 Linux 工具

Linux工具	备注
which	shows the full path of (shell) commands.
update-rc.d,chkconfig	updates and queries runlevel information for system services
test	check file types and compare values
groups	print the groups a user is in
awk	pattern scanning and processing language
chattr	change file attributes on a Linux file system
lsattr	list file attributes on a Linux second extended file system

Linux工具	备注
ldconfig	configure dynamic linker run-time bindings
wc	print newline, word, and byte counts for each file
lspci	list all PCI devices
ifconfig	configure a network interface
chown	change file owner and group
chmod	change file mode bits
find	search for files in a directory hierarchy
sed	stream editor for filtering and transforming text
ln	make links between files
modprobe	Add and remove modules from the Linux Kernel
cksum	checksum and count the bytes in a file
tail	output the last part of files
head	output the first part of files
udevadm	udev management tool
modinfo	show information about a Linux Kernel module
useradd	create a new user or update default new user information
expr	evaluate expressions
basename	strip directory and suffix from filenames
tty	print the file name of the terminal connected to standard input
dirname	strip last component from file name
rev	reverse lines characterwise
id	print real and effective user and group IDs
touch	change file timestamps
grep	print lines matching a pattern
ps	report a snapshot of the current processes
useradd	create a new user or update default new user information

8.5.3 相关工具

须知

upgrade-tool工具不支持容器场景，部署容器时，禁止将upgrade-tool工具映射到容器内。

步骤1 以安装软件包时指定的运行用户（默认HwHiAiUser）登录Host侧的服务器。

步骤2 切换到软件包所在安装路径，根据以下表格中的命令调用工具，如下以Driver采用默认安装路径为例进行说明。

版权所有 © 华为技术有限公司

表 8-9 相关工具

名称	路径	作用	执行命令	参数
upgrade-tool	/usr/local/Ascend/driver/tools	查看固件版本、升级单个或多个固件等，只允许root用户执行。	请切换到/usr/local/Ascend/driver/tools路径执行如下命令： ● 列举所有Device：**./upgrade-tool --mini_devices** ● 获取指定设备的版本：**./upgrade-tool --device_index <dev_id> --system_version** ● 获取指定设备的组件信息：**./upgrade-tool --device_index <dev_id> --components** ● 查询某个设备中某一个组件对应的版本：**./upgrade-tool --device_index <dev_id>--component <type> --version** ● 查询设备状态：	● --mini_devices：所有设备列表。 ● --device_index：设备编号。取值可以是【0~63】和-1，【0~63】表示对应编号的设备。-1表示所有设备。 ● --system_version：系统版本。 ● --components：列举所有有效的组件。 ● --component：指定具体的组件。升级单个组件时，需要指定组件名称。升级所有组件时，需要输入-1。需要升级所有组件并清除用户配置数据时，请输入-9。 ● --version：组件的版本。 ● --status：设备状态。支持的设备

版权所有 © 华为技术有限公司

名称	路径	作用	执行命令	参数
			./upgrade-tool --device_index <dev_id> --status ● 查询设备是否是物理机：**./upgrade-tool --device_index <dev_id> --phymachflag** **说明** 仅支持通过物理机升级固件包。 ● 启动固件主备同步：**./upgrade-tool --sync** ● 查看所有分区的固件版本：**./upgrade-tool --device_index <dev_id> --component <type> --all --version** ● 配置算力功率等级：**./upgrade-tool --device_index <dev_id> --component nve --level**	状态具体包含： – idle：空闲。 – upgrading：正在升级。 – not support：不支持。 – failed：失败。 – waitting_restart：等待重启。 – waitting_sync：等待固件同步。 – synchronizing：正在同步。 – wrong status：错误状态。 ● --phymachflag：查询设备是否是物理机。如果不是物理机，不允许升级设备的固件包。 ● --async：支持异步升级，即device侧收到host侧发送的升级请求后，返回

名称	路径	作用	执行命令	参数
			<low/middle/high/full> ● 指定组件的主备区域进行升级：**./upgrade-tool --device_index <dev_id>--component <type> --media --mmc <main/backup> --path <firmware_path>** ● 查询主备分区的使能状态：**./upgrade-tool --fs_backup_status** ● 获取当前设备的启动状态：**./upgrade-tool --device_index <dev_id> --boot_status** ● 设置当前设备热复位标志：**./upgrade-tool --device_index <dev_id> --hot_reset_flag**	请求成功响应到host侧（具体固件是否升级成功，需要通过“--status”参数去查询）。 ● --help：查看帮助信息。 ● --sync：启动固件主备同步，仅支持RC场景。 ● --all --version：查看所有分区（Flash分区、主备分区）的固件版本，仅支持RC场景。 ● --level：配置算力功率等级，仅支持RC场景。 ● --media --mmc：指定组件的主备区域进行升级，仅支持RC场景。 ● --fs_backup_status：主备分区的使能状态，仅支持RC场景。 ● --boot_status：当前设

名称	路径	作用	执行命令	参数
				备的启动状态。 – 0：未初始化。 – 1：BIOS启动中。 – 2：OS启动中。 – 3：启动完成。 – 16：DSMI服务进程启动完成。 ● --hot_reset_flag：当前设备的热复位标志，不支持RC场景。

名称	路径	作用	执行命令	参数
			请切换到/usr/local/Ascend/firmware/tools路径执行如下命令： ● 升级指定设备的固件包：**/usr/local/Ascend/driver/tools/upgrade-tool --device_index <dev_id> --component <type> --path <firmware_path>** ● 异步升级指定设备的固件包：**/usr/local/Ascend/driver/tools/upgrade-tool --device_index <dev_id> --component <type> --async --path <firmware_path>** **--async** 必须紧跟在**component <type>**后面。	--path： ● 固件包相对路径：若升级全部组件，则为**--path ./conf/upgrade.cfg**；“upgrade.cfg”文件中包含各个组件的相对路径。若升级单个组件，例如nve.bin，则为**--path ../image/nve.bin**。

----结束

版权所有 © 华为技术有限公司

9 安装与维护（适用于 1.0.0-1.0.10 版本）

Atlas 300I 推理卡（型号 3000）是标准的PCIe卡，其硬件安装方式与常规PCIe卡相同，安装方法可参见各服务器用户指南。此处仅介绍Atlas 300I 推理卡（型号 3000）驱动和固件的安装与维护。

9.1 安装流程

9.2 确认操作系统

9.3 获取软件包

9.4 创建运行用户

9.5 安装驱动

9.6 安装固件

9.7 卸载驱动

9.8 卸载固件

9.9 常用操作

9.10 常见问题

9.11 参考

9.1 安装流程

安装流程

图 9-1 Atlas 300I 推理卡（型号 3000）软件包安装流程

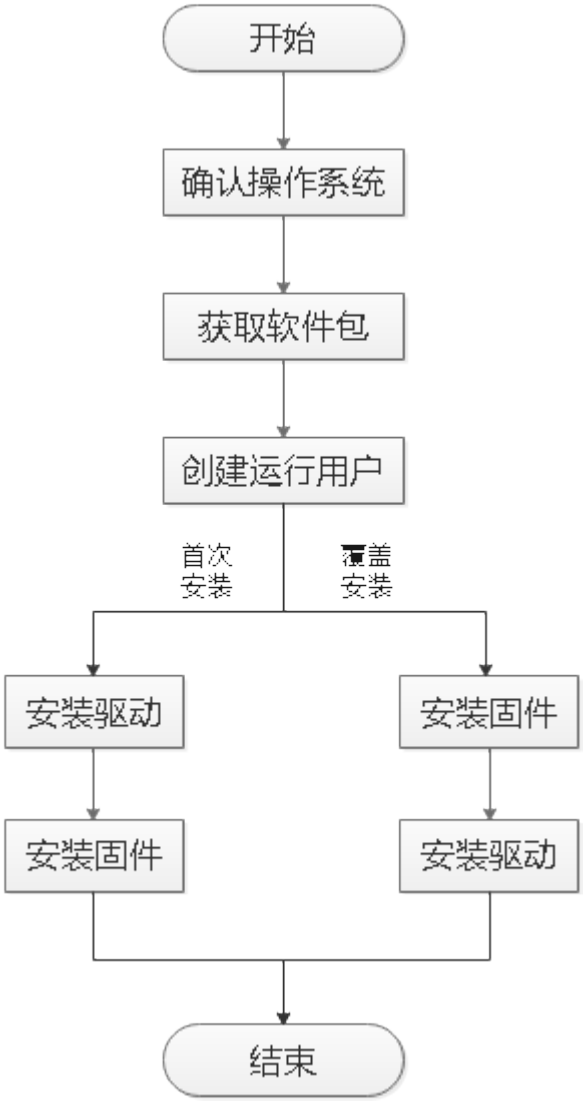


表 9-1 安装流程说明

关键步骤	说明	参考章节
确认操作系统	确认目标环境支持的操作系统。	9.2 确认操作系统
获取软件包	根据目标环境所使用的操作系统下载所需软件包。	9.3 获取软件包

版权所有 © 华为技术有限公司

关键步骤	说明	参考章节
创建运行用户	用户安装软件包时使用root用户，但安装完之后要求使用非root用户运行，所以安装前需要先创建运行用户。	9.4 创建运行用户
安装驱动	支持以下三种安装场景。 ● 二进制文件直接安装 ● 源码编译安装 ● 重构驱动包安装 **说明** ● 使用NPU 1.X.X系列版本软件包安装时，固件包集成在驱动包内，安装或升级驱动过程中会自动安装和升级固件。 ● 使用NPU 20.X.X及以上系列版本软件包安装时，首次安装请按照“**驱动->固件**”的顺序；覆盖安装或升级请按照“**固件->驱动**”的顺序，分别安装软件包。	9.5 安装驱动
安装固件	根据不同的软件包格式指导用户安装固件。 **说明** ● 使用NPU 1.X.X系列版本软件包安装时，固件包集成在驱动包内，安装或升级驱动过程中会自动安装和升级固件。 ● 使用NPU 20.X.X及以上系列版本软件包安装时，首次安装请按照“**驱动->固件**”的顺序；覆盖安装或升级请按照“**固件->驱动**”的顺序，分别安装软件包。	9.6 安装固件

9.2 确认操作系统

操作步骤

步骤1 查询服务器支持的操作系统，详细请参考计算产品兼容性查询助手。

步骤2 执行如下命令，查询服务器当前运行环境的操作系统架构及版本。

```
uname -m && cat /etc/*release
```

说明

如果查询的操作系统版本不在兼容性列表中，请替换为兼容性列表中支持的操作系统。

----结束

版权所有 © 华为技术有限公司

9.3 获取软件包

操作步骤

- 企业网用户：
  a. 登录**A300-3000**的软件页签。
  b. 打开目标版本“A300-3000 *X*”。
  c. 查看《版本配套表》，选择所需的软件包。
  d. 单击软件包（如A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run）后面的和，可获取软件包和数字签名文件。

  说明

  请获取同一版本下的驱动包、芯片固件包、MCU固件包以及驱动包重新打包工具。
- 运营商用户：联系华为技术有限公司驻当地办事处的技术支持人员。

9.3.1 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：**http://support.huawei.com/carrier/digitalSignatureAction**

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

9.4 创建运行用户

安装驱动后要求使用非root用户运行相关进程，安装前则需要先创建运行用户。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

表 9-2 创建运行用户

软件包版本	运行用户	操作步骤
NPU 21.0.X	• HwHiAiUser • 非HwHiAiUser	• 如果创建的用户是HwHiAiUser，安装软件包时可以直接使用该运行用户，默认即为HwHiAiUser。rpm
NPU 20.2.X		

版权所有 © 华为技术有限公司

软件包版本	运行用户	操作步骤
NPU 20.1.X	通过**--install-username=*username* --install-usergroup=*usergroup***参数指定用户，推荐指定非root用户；创建的该用户不能为root用户属组。	包安装过程中，若无HwHiAiUser用户，系统会自动创建该用户。 ● 如果创建的用户是非HwHiAiUser，安装驱动软件包时需要指定运行用户（通过**--install-username=***username* **--install-usergroup=***usergroup*参数指定），请参见如下方法创建运行用户。 1. 登录Atlas 300I 推理卡（型号3000）所在服务器，并在root用户下执行以下操作。 2. 创建运行用户。 **# groupadd** *usergroup* **# useradd -g** *usergroup* **-d /home/***username* **-m** *username* **-s /bin/bash**
NPU 20.0.X NPU 1.X.X	HwHiAiUser	1. 登录Atlas 300I 推理卡（型号3000）所在服务器，并在root用户下执行以下操作。 2. 创建运行用户。 **# groupadd HwHiAiUser** **# useradd -g HwHiAiUser -d /home/HwHiAiUser -m HwHiAiUser -s /bin/bash**

9.5 安装驱动

9.5.1 确认安装方式

请根据获取的驱动包类型以及安装场景说明选择驱动安装方式。

表 9-3 安装场景说明

驱动包类型	安装场景	安装场景说明	安装方式
二进制驱动包	二进制文件直接安装	该方式适用于《版本配套表》中驱动包默认内核版本和运行环境内核版本一致的场景。	请参见**9.5.2 二进制文件直接安装**。
	源码编译安装	该方式适用于《版本配套表》中驱动包默认内核版本和运行环境内核版本不一致的场景。	请参见**9.5.3 源码编译安装**。

版权所有 © 华为技术有限公司

驱动包类型	安装场景	安装场景说明	安装方式
源码驱动包	源码编译安装	该方式适用于软件包无须批量安装的场景。	请参见**9.5.3 源码编译安装**。
	重构驱动包安装	该方式适用于软件包批量离线安装的场景	请参考**9.5.4 重构驱动包安装**

系统兼容性要求

- 支持PCIe MSI-X中断，每个Atlas 300I 推理卡至少可分配464个MSI-X中断。
- BAR空间大小要求：
  Atlas 300I 推理卡需要三个BAR，空间大小分别为512KB，64MB，256MB。

环境检查

表 9-4 环境检查

序号	检查项	检查命令	检查说明
1	操作系统内核版本	**uname -r**	• 若当前运行环境内核版本和《版本配套表》中操作系统默认内核版本一致，安装驱动请参见**9.5.2 二进制文件直接安装**。。 • 若当前运行环境内核版本和《版本配套表》中操作系统默认内核版本不一致，用户可基于安装包驱动源码重新构建安装使用，具体请参见**9.5.3 源码编译安装**。 **说明** • 如果Host系统中安装过run包，内核升级后会出现run包启动异常的问题。出现该问题时请参见**9.10.4 升级系统内核版本时未卸载run包导致run包不可用**解决。 • 对于EulerOS-Virtual-V200R009C10操作系统，当前host操作系统的内核版本必须与《版本配套表》中要求一致。 • EulerOS 2.8场景下，操作系统内核版本必须与《版本配套表》中默认系统版本一致；如果不一致，只能通过升级内核版本的方式解决。否则会导致安装过程报错：部分芯片无法启动。 • 安装驱动过程中出现：“driver ko vermagic(*) is different from the os(*), need rebuild driver”，请参见**9.5.3 源码编译安装**解决。
2	GCC版本	**gcc -v**	• 若使用二进制文件直接安装驱动包，对于驱动包名称中包含gcc版本的，要求用户的gcc版本必须与驱动包名称中的gcc版本保持一致；驱动包名称中不包含gcc版本的，安装无须检查GCC版本。 • 若使用源码编译安装驱动包，GCC版本须不低于《版本配套表》中要求的版本。

版权所有 © 华为技术有限公司

序号	检查项	检查命令	检查说明
3	查询系统是否安装过run包	**lsmod| grep drv_pcie_host**	• 如无内容表示未安装过run包。可以直接安装软件包。 • 如有内容，表示安装过run包。需要先卸载run包后，再安装新版本软件包。
4	检测Atlas 300I 推理卡（型号 3000）是否正常在位	**lspci | grep d100**	• 如果服务器上有*N*（*N*>0）张Atlas 300I 推理卡（型号 3000），回显中含“d100”字段的行数为4*N*，则表示Atlas 300I 推理卡（型号 3000）正常在位。 88:00.0 Processing accelerators: Huawei Technologies Co., Ltd. Device d100 (rev 20) 89:00.0 Processing accelerators: Huawei Technologies Co., Ltd. Device d100 (rev 20) 8a:00.0 Processing accelerators: Huawei Technologies Co., Ltd. Device d100 (rev 20) 8b:00.0 Processing accelerators: Huawei Technologies Co., Ltd. Device d100 (rev 20) ..... • 如果Atlas 300I 推理卡（型号 3000）所在的服务器是华为服务器，可通过登录iBMC WebUI界面，进入“信息管理 > 系统信息 ”，单击“其他”，Atlas 300I 推理卡（型号 3000）的PCIe卡信息在“PCIe卡”列表中，表示Atlas 300I 推理卡（型号 3000）正常在位。

9.5.2 二进制文件直接安装

9.5.2.1 安装驱动（适用于.run 格式）

Atlas 300I 推理卡（型号 3000）发布的run安装包是host安装包，用户下载和解压安装包需要在host操作系统完成。本章节介绍宿主机上的安装方法，容器内安装请参见12 容器内安装。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

注意事项

- run安装包支持一键式安装，用户根据安装需求选择对应参数完成安装，具体请参见9.11.1 参数说明。
- 如果有安装NPU 1.X.X系列版本的驱动包，在安装NPU 20.X.X及以上系列版本的驱动包前需要先卸载低版本驱动。
- 使用run安装包后，不要手动设置环境变量export LD_LIBRARY_PATH指向之前驱动包的SO文件，否则可能会出现run安装包内工具连接到之前版本的动态库。指向第三方库文件路径、非run安装包发布库文件路径的配置不受影响。

- 在软件包安装或升级过程中，不允许对Host或Device进行复位或下电操作，否则会出现Device无法正常启动或者升级失败。若出现此问题，解决方法请参见**9.10.13 复位或下电导致Device无法正常启动**。

安装 NPU 20.X.X 及以上系列版本驱动包

以*A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run*为例介绍.run格式驱动包安装相关指导，具体请根据实际情况进行替换。

须知

首次安装请按照“**驱动->固件**”的顺序；覆盖安装或升级请按照“**固件->驱动**”的顺序，分别安装软件包。固件安装请参见**9.6.1 安装固件（适用于.run格式）**。

为保证驱动版本、固件版本和MCU版本三者保持配套关系，请参考《**Atlas 300I 推理卡 NPU驱动和固件升级指南（型号 3000, 3010）**》中“升级MCU”章节升级对应的MCU组件。

步骤1 将**9.3 获取软件包**准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见**9.9.5 通过PuTTY登录（网口方式）**。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 执行如下命令，进入软件包所在路径（如“/opt”）。

cd */opt*

步骤5 执行如下命令，增加软件包的可执行权限。

chmod +x *A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run*

步骤6 执行如下命令，校验run安装包的一致性和完整性。

./*A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run* **--check**

若出现如下回显信息，表示软件包校验成功。

```
Verifying archive integrity...  100%   SHA256 checksums are OK. All good.
```

说明

软件包校验使用SHA256加密方式；校验过程中若提示：./*A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run* does not contain an embedded MD5 checksum.和./*A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run* does not contain a CRC checksum，表示没有使用MD5和CRC加密方式，可忽略。

步骤7 执行安装命令完成安装。

- 若用户指定安装路径，以“/test/HiAI/”为例。

  可执行**./***A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run* **--full --install-path=** /test/HiAI/命令，完成安装。

- 若未指定安装路径，可执行**./***A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run* **--full**命令，完成安装。

说明

- 安装run包驱动时，会将动态库libdcmi.so和头文件dcmi_interface_api.h拷贝到“/usr/local/dcmi/”目录下。
- run安装包安装过程中出现的异常场景及其解决方法请参见**9.10 常见问题**。
- 若创建运行用户是非HwHiAiUser，安装驱动软件包时需要指定运行用户（通过**--install-username=***username* **--install-usergroup=***usergroup*参数指定）。
- 若用户指定安装路径安装驱动时：
  - 若指定路径不存在，则安装时会自动创建目录，若有多层目录，则只有最后一层目录不存在时会自动创建。
  - 若指定路径已存在：
    - 若该路径下所有层级目录属主为root用户，则请确保所有层级目录权限至少为755。
    - 若该路径下有一层目录属主为非root用户，则请用户自行修改为root属主，并确保所有层级目录权限为755。
- 软件包默认安装路径：/usr/local/Ascend
- 安装详细日志路径：/var/log/ascend_seclog/ascend_install.log。
- 安装后软件包的安装路径、安装命令以及运行用户信息记录路径：/etc/ascend_install.info
- 使用NPU 20.0.X和NPU 20.1.X系列版本驱动包安装时，以非HwHiAiUser用户为运行用户安装驱动，卸载重启后以HwHiAiUser为运行用户再次安装，若出现“/home/HwHiAiUser: Permission denied”报错，请忽略。
- 使用NPU 20.2.X及以上系列版本软件包安装时，Device侧系统类日志通过msnpureport工具传输到Host侧进行查看。导出操作以及导出日志的存储路径具体请参见**《Ascend 310 黑匣子日志参考》**附录中的“msnpureport工具使用”章节。

  容器内不支持查看Device侧系统类日志，也不支持通过msnpureport工具导出Device侧系统类日志。

步骤8 若系统显示安装成功提示信息，则表示驱动已安装成功。根据系统提示信息决定是否重启系统，若需要重启，请执行以下操作；否则，请跳过此步骤。

1. 安装完成后，在host操作系统执行**reboot**命令复位，或将host机下电再上电。
2. host复位后，自动安装驱动及服务，Ascend 310自动进入操作系统。

步骤9 执行**npu-smi info**查看驱动加载是否成功。

若出现如下回显信息，说明加载成功。否则，说明安装失败，请联系华为技术支持处理。

```
+------------------------------------------------------------------------------+
| npu-smi 20.0.1                    Version: 1.73.5.10.050                     |
+-------------------+-----------------+-------------------------------------+
| NPU     Name      | Health          | Power(W)          Temp(C)           |
| Chip    Device    | Bus-Id          | AICore(%)         Memory-Usage(MB)  |
+===================+=================+=====================================+
| 2049    310       | OK              | 12.8              50                |
| 0       0         | 0000:85:00.0    | 0                 2621 / 8192       |
+-------------------+-----------------+-------------------------------------+
| 2049    310       | OK              | 12.8              51                |
| 1       1         | 0000:86:00.0    | 0                 2621 / 8192       |
+-------------------+-----------------+-------------------------------------+
| 2049    310       | OK              | 12.8              52                |
| 2       2         | 0000:87:00.0    | 0                 2621 / 8192       |
+-------------------+-----------------+-------------------------------------+
| 2049    310       | OK              | 12.8              55                |
| 3       3         | 0000:88:00.0    | 0                 2621 / 8192       |
+===================+=================+=====================================+
```


版权所有 © 华为技术有限公司

说明

回显中“npu-smi”后面的字段为npu-smi工具版本号；“Version:”后面的字段为驱动版本号。

----结束

后续处理

使用NPU 20.2.X及以上系列版本软件包安装时，请跳过此步骤。使用NPU 20.0.X和NPU 20.1.X系列版本软件包安装时，ADA默认开放侦听端口22118。为确保系统安全，在商用环境中，需要关闭该端口。

须知

1. “/usr/local/Ascend/driver”为使用root用户安装驱动（add-ons随驱动安装）的默认安装路径，下文以此为例来说明操作步骤，请根据实际情况替换。
2. 以root用户安装运行环境时，HwHiAiUser用户为运行环境上的默认进程运行用户，下文以此为例来说明操作步骤。

1. 使用**su - HwHiAiUser**命令切换到运行用户，登录运行环境，打开文件“/home/HwHiAiUser/ide_daemon/ide_daemon.cfg”。
   - 如果SOCK_SWITCH=0，则表示ADA端口已关闭。
   - 如果SOCK_SWITCH=1，则表示ADA端口为开启状态，需要用户手动修改为SOCK_SWITCH=0。

   修改完后执行**wq!**保存退出。
2. 切换到“/usr/local/Ascend/driver/tools”目录下，依次执行如下命令设置环境变量、重启ada进程或执行**reboot**命令重启运行环境。
   a. 设置环境变量。

      export LD_LIBRARY_PATH=/usr/local/Ascend/add-ons:$LD_LIBRARY_PATH
   b. 查找ada进程的ID。

      ps -ef | grep ada
   c. 停止ada进程。

      kill -9 *ada进程ID*
   d. 启动ada进程。

      如果执行如下命令时，提示“ada is exist, don't start again”，则表示系统已自动拉起ada进程，无需再手动执行启动命令。

      /usr/local/Ascend/driver/tools/ada

安装 NPU 1.X.X 系列版本驱动包

以A800-3000-NPU-Ubuntu18.04-ARM-V*x.x.x*.run为例介绍.run格式驱动包安装相关指导，具体请根据实际情况进行替换。

须知

目前仅支持full模式进行安装，即全包安装模式。

版权所有 © 华为技术有限公司

步骤1 将9.3 获取软件包准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见9.9.5 通过PuTTY登录（网口方式）。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 执行如下命令，进入软件包所在路径（如“/opt”）。

cd */opt*

步骤5 执行如下命令，增加软件包的可执行权限。

chmod +x *A800-3000-NPU-Ubuntu18.04-ARM-Vx.x.x.run*

步骤6 执行如下命令，校验run安装包的一致性和完整性。

./*A800-3000-NPU-Ubuntu18.04-ARM-Vx.x.x.run* **--check**

若出现如下回显信息，表示软件包校验成功。

```
Verifying archive integrity...  100%   SHA256 checksums are OK. All good.
```

步骤7 设置安装路径。

- 如果用户需要指定安装路径，则需要先创建安装路径。以“/test/HiAI/”为例，用户先执行**mkdir /test/HiAI/**命令，完成安装路径的创建，再选择该安装路径进行软件安装。
- 如果用户不需要指定安装路径，安装包会安装到默认路径“/usr/local/HiAI”下。

步骤8 执行安装命令完成安装。

安装过程支持一键式安装操作。

- 若用户指定安装路径，以“/test/HiAI/”为例。

  可执行**./***A800-3000-NPU-Ubuntu18.04-ARM-Vx.x.x.run* **--full --path /test/HiAI/**命令，完成安装。
- 若未指定安装路径，可执行**./***A800-3000-NPU-Ubuntu18.04-ARM-Vx.x.x.run* **--full**命令，完成安装。

说明

- 安装run包驱动时，会将动态库libdcmi.so和头文件dcmi_interface_api.h拷贝到“/usr/local/dcmi/”目录下。
- run安装包安装过程中出现的异常场景及其解决方法请参见9.10 常见问题。
- 安装过程中Ascend 310驱动、固件安装和升级的日志信息输出在“/var/log/HiAI-install.log”文件中，用户可执行**./****.run* **--viewlog**命令，打开日志；npu-smi工具安装的日志信息输出在“/var/log/it_logs”中。
- 安装驱动过程中出现“drv_dkms_env_check failed, details in : /var/log/HiAI-install.log”的报错，请查看日志信息，日志中显示“The program 'dkms' is currently not installed.”，说明错误为缺少DKMS，请安装DKMS。具体操作请参见9.10.3 安装驱动过程中出现“drv_dkms_env_check failed, details in : /var/log/HiAI-install.log”的报错。

步骤9 run安装包安装完成后需要复位host主机，复位后，运行的Ascend 310软件和mini固件即为安装的最新版本。复位host主机步骤如下。

1. 安装完成后，在host操作系统执行**reboot**命令复位，或将host机下电再上电。
2. host复位后，自动安装驱动及服务，Ascend 310自动进入操作系统。

步骤10 执行**npu-smi info**查看驱动加载是否成功。

若出现如下回显信息，说明加载成功。否则，说明加载失败，请联系华为技术支持处理。

```
root@atlas-G560:~# npu-smi info
+------------------------------------------------------------------------------+
| npu-smi 1.1.2                    Version: 1.1.T19.B880                       |
+-------------------+-----------------+-------------------------------------+
| NPU     Name      | Health          | Power(W)          Temp(C)           |
| Chip              | Bus-Id          | AICore(%)         Memory-Usage(MB)  |
+===================+=================+=====================================+
| 77      HI1910    | Warning         | 12.8              61                |
| 0                 | 0000:4F:00.0    | 0                 2457 / 8192       |
+-------------------+-----------------+-------------------------------------+
| 77      HI1910    | OK              | 12.8              58                |
| 1                 | 0000:53:00.0    | 0                 2375 / 8192       |
+-------------------+-----------------+-------------------------------------+
| 77      HI1910    | OK              | 12.8              57                |
| 2                 | 0000:57:00.0    | 0                 2375 / 8192       |
+-------------------+-----------------+-------------------------------------+
| 77      HI1910    | OK              | 12.8              58                |
| 3                 | 0000:5B:00.0    | 0                 2375 / 8192       |
+===================+=================+=====================================+
```

说明

回显中“npu-smi”后面的字段为npu-smi工具版本号，Version:”后面的字段为驱动/芯片固件版本号。

----结束

9.5.2.2 安装驱动（适用于.rpm 格式）

Atlas 300I 推理卡（型号 3000）发布的.rpm格式安装包是host安装包，用户下载和解压安装包需要在host操作系统完成。如果rpm安装包已经安装，则可以选择升级命令进行升级。升级操作可根据具体的升级版本参见对应的《**Atlas 300I 推理卡 NPU驱动和固件升级指南（型号 3000, 3010）**》。

注意事项

- 当前rpm包仅适配EulerOS系统，其他环境请不要使用。
- 当前rpm包安装不支持修改安装路径功能，请按照正常rpm流程安装，不支持rpm包的--force命令安装升级。
- rpm包与run包为同级别组件，两者不能兼容，即安装run包后不能再安装rpm包、安装rpm包后也不支持再安装run包。
- Ascend 310内部部分功能，只支持HwHiAiUser用户访问和执行，用户首次安装时系统会自动创建HwHiAiUser用户。
- 安装完成后需要重启生效。卸载完成后，如需再次安装，需要先重启再安装。
- rpm包的驱动升级功能不支持回退，如需回退需先删除驱动，再重新安装低版本驱动。
- 如果已安装NPU 1.X.X系列版本的rpm包，在安装NPU 20.X.X及以上系列版本的rpm包前需要先卸载原版本再安装。
- 在软件包安装或升级过程中，不允许对Host或Device进行复位或下电操作，否则会出现Device无法正常启动或者升级失败。若出现此问题，解决方法请参见**9.10.13 复位或下电导致Device无法正常启动**。

安装 NPU 20.X.X 及以上系列版本驱动包

以A300-3000-npu-driver_*x.x.x*_euleros-virtual2.9-aarch64.rpm为例介绍.rpm格式驱动包安装相关指导，具体请根据实际情况进行替换。

须知

npu-smi工具集成在驱动包内，安装驱动过程中会自动安装npu-smi工具。固件包支持单独安装，安装请按照“**驱动->固件**”的顺序，分别安装组件rpm包。固件安装请参见**9.6.2 安装固件（适用于.rpm格式）**。

步骤1 将**9.3 获取软件包**准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见**9.9.5 通过PuTTY登录（网口方式）**。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 在软件包所在路径执行如下命令完成Host侧驱动包安装。

rpm -ivh *A300-3000-npu-driver_x.x.x_euleros-virtual2.9-aarch64.rpm*

说明

软件包安装过程中，若无HwHiAiUser用户系统会自动创建该用户。

步骤5 执行如下命令，重启Host侧系统使驱动生效。

reboot

步骤6 验证驱动安装是否成功。

1. 执行如下命令，查看Atlas 300I 推理卡（型号 3000）是否正常在位。

   lspci | grep d100

   如果服务器上有*N*（*N*>0）张Atlas 300I 推理卡（型号 3000），回显中含“d100”字段的行数为4*N*，则表示Atlas 300I 推理卡（型号 3000）正常在位。

```
91:00.0 Processing accelerators: Huawei Technologies Co., Ltd. Device d100 (rev 20)
92:00.0 Processing accelerators: Huawei Technologies Co., Ltd. Device d100 (rev 20)
93:00.0 Processing accelerators: Huawei Technologies Co., Ltd. Device d100 (rev 20)
94:00.0 Processing accelerators: Huawei Technologies Co., Ltd. Device d100 (rev 20)
```

2. 执行如下命令，查看驱动安装是否成功。

   npu-smi info

   回显如下类似信息，说明安装成功。否则，说明安装失败，请联系华为工程师处理。

```
+------------------------------------------------------------------------------+
| npu-smi 20.2.0                   Version: 20.2.0                             |
+-------------------+-----------------+-----------------------------------------+
| NPU     Name      | Health          | Power(W)          Temp(C)               |
| Chip    Device    | Bus-Id          | AICore(%)         Memory-Usage(MB)      |
+===================+=================+=========================================+
| 215     310       | OK              | 12.8              56                    |
| 0       0         | 0000:DA:00.0    | 0                 2703 / 8192           |
+-------------------+-----------------+-----------------------------------------+
| 215     310       | OK              | 12.8              57                    |
| 1       1         | 0000:DC:00.0    | 0                 2703 / 8192           |
+-------------------+-----------------+-----------------------------------------+
```


版权所有 © 华为技术有限公司

```
| 215     310       | OK            | 12.8            57                |
| 2       2         | 0000:DD:00.0  | 0               2703 / 8192       |
+-------------------+---------------+-----------------------------------+
| 215     310       | OK            | 12.8            55                |
| 3       3         | 0000:DB:00.0  | 0               2703 / 8192       |
+===================+===============+===================================+
```

说明

安装过程中的日志信息输出在“/var/log/ascend_seclog/ascend_install.log”文件中，用户可以执行**vim /var/log/ascend_seclog/ascend_install.log**命令打开日志。

----结束

安装 NPU 1.X.X 系列版本驱动包

以HiAI_Euler_V2R8_driver-<version>-1.aarch64.rpm为例介绍.rpm格式驱动包安装相关指导，具体请根据实际情况进行替换。

须知

固件包集成在驱动包内，安装或升级驱动过程中会自动升级固件。npu-smi工具支持单独安装，具体请参见**10.1.1 安装npu-smi工具**。

步骤1 按照**9.3 获取软件包**准备软件压缩包“A800-3000-NPU-EulerOS2.8-ARM-V*x.x.x*.zip”。

步骤2 解压“A800-3000-NPU-EulerOS2.8-ARM-V*x.x.x*.zip”rpm安装包“HiAI_Euler_V2R8_driver-*<version>*-1.aarch64.rpm”。

步骤3 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见**9.9.5 通过PuTTY登录（网口方式）**。

步骤4 执行如下命令，切换至root用户。

su - root

步骤5 将“HiAI_Euler_V2R8_driver-*<version>*-1.aarch64.rpm”安装包上传至Linux系统任意目录下，如/opt下。

步骤6 执行**rpm -ivh** *HiAI_Euler_V2R8_driver-<version>-1.aarch64.rpm*命令完成rpm包安装。

步骤7 执行**reboot**命令重启系统使驱动生效。

步骤8 执行**/usr/local/HiAI/firmware/upgrade-tool --device_index -1 --component -1 --version**查看驱动安装是否成功。

回显如下信息，说明安装成功。否则，说明安装失败，请联系华为技术支持处理。

```
{"device_id"=0, "component":nev, "version":1.3.10.770}
{"device_id"=0, "component":xloader, "version":1.3.10.770}
{"device_id"=0, "component":m3fw, "version":1.3.10.770}
{"device_id"=0, "component":uefi, "version":1.3.10.770}
{"device_id"=0, "component":kernel, "version":1.3.10.770}
{"device_id"=0, "component":dtb, "version":1.3.10.770}
```


版权所有 © 华为技术有限公司

说明

安装过程中的日志信息输出在“/var/log/HiAI-install.log”文件中，用户可以执行**vim /var/log/HiAI-install.log**命令打开日志。

----结束

9.5.3 源码编译安装

当用户当前操作系统内核版本更新或与默认不一致，可以基于安装包驱动源码重新构建并安装使用，该场景下不会触发内核版本升级，会基于用户当前内核版本编译出相应的内核驱动。软件包在安装时会自动触发驱动源码编译，编译完成后自动进行软件包安装。

如果使用该功能，则需要提前检查并安装相关依赖工具。具体操作请切换到root用户执行。

9.5.3.1 安装依赖

源码驱动包特有安装依赖

- make工具

  执行**make -v**命令，若能查询到make工具的版本，则make已安装。
- 默认内核源码路径：例如：/lib/modules/`uname -r`/build

  执行**ls /lib/modules/`uname -r`/build**命令，查看路径是否存在。
  - 若存在，安装通用软件包时则会自动使用内核进行驱动编译。
  - 若不存在，您可以在安装时提供对应的源码路径，具体请参见**步骤8**。

  或dkms等安装依赖，具体请参见以下各操作系统。对于**Kylinosv10**、**UOS 20**、**kylinosv10SP1**、**NeoKylin 7.6**、**Tlinux 2.4**、**Linx 6.0.90**操作系统，请忽略dkms依赖检查。

  说明

  以下操作系统中的DKMS软件包建议使用dkms-2.6.1-1.el7.noarch.rpm。

Ubuntu 18.04 操作系统

需要安装dkms、gcc、linux-header软件包。执行如下命令检查：

dpkg-query -s dkms

dpkg-query -s gcc

dpkg-query -s linux-headers-$(uname -r)

若未安装请自行在相应操作系统厂商网站获取并安装。

CentOS 7.6/8.2 操作系统

需要安装dkms、gcc、kernel-headers、kernel-devel软件包，执行如下命令检查：

rpm -qa | grep dkms

rpm -qa | grep gcc

版权所有 © 华为技术有限公司

rpm -qa | grep kernel-headers-$(uname -r)

rpm -qa | grep kernel-devel-$(uname -r)

若未安装请自行在相应操作系统厂商网站获取并安装。

EulerOS 2.8 操作系统

需要安装gcc、kernel-headers、kernel-devel软件包。执行如下命令检查：

rpm -qa | grep gcc

rpm -qa | grep kernel-headers-$(uname -r)

rpm -qa | grep kernel-devel-$(uname -r)

若未安装请自行在相应操作系统厂商网站获取并安装。

SUSE12SP5 操作系统

1. 需要安装gcc、kernel-default、kernel-default-devel软件包。执行如下命令检查：

   rpm -qa | grep gcc

   rpm -qa | grep kernel-default

   rpm -qa | grep kernel-default-devel

   若未安装请自行在相应操作系统厂商网站获取并安装。

2. 检查配置文件“10-unsupported-modules.conf”中“allow_unsupported_modules ”的值是否为1。

   执行**cat /etc/modprobe.d/10-unsupported-modules.conf**命令，查看modprobe限制配置文件中“allow_unsupported_modules ”的值是否“1”。

   如果不是，使用**vi /etc/modprobe.d/10-unsupported-modules.conf**命令修改“allow_unsupported_modules ”的值为“1”。

   说明

   在SUSE12SP5操作系统下，可通过“10-unsupported-modules.conf”配置文件控制非系统自带驱动在系统启动过程中的加载。默认情况下，“10-unsupported-modules.conf”配置文件中“allow_unsupported_modules ”的值为“0”，表示不允许系统启动过程中加载非系统自带驱动；“allow_unsupported_modules ”的值设置为“1”时，将允许系统启动过程中加载非系统自带驱动。

Linx 6.0.90 操作系统

检查“/etc/pam.d/su”配置文件的“auth sufficient pam_rootok.so”字段是否被注释掉，若被注释，请参见**9.10.14 root用户切换到普通用户需要输入密码权限**解决。

EulerOS 2.9 操作系统

需要安装kernel-headers、gcc 7.3.0、elfutils-devel、make等软件包。操作如下：

1. 请获取对应版本OS的“devel_tools.tar.gz”工具组件。
2. 上传“devel_tools.tar.gz”到运行环境的任一目录。如：/opt下。
3. 进入“devel_tools.tar.gz”所在路径并进行解压。

   cd /opt

tar -xzvf devel_tools.tar.gz

4. 在3的解压目录下，执行以下命令安装kernel-headers、gcc 7.3.0、elfutils-devel、make等软件包。
   - 对于**EulerOS 2.9+aarch64**系统，请执行如下命令：

     rpm -ivh kernel-headers-4.19.90-vhulk2009.2.0.h269.eulerosv2r9.aarch64.rpm kernel-devel-4.19.90-vhulk2009.2.0.h269.eulerosv2r9.aarch64.rpm

     rpm -ivh binutils-2.34-1.h11.eulerosv2r9.aarch64.rpm binutils-extra-2.34-1.h11.eulerosv2r9.aarch64.rpm binutils-devel-2.34-1.h11.eulerosv2r9.aarch64.rpm zlib-devel-1.2.11-17.eulerosv2r9.aarch64.rpm cpp-7.3.0-20190804.h39.eulerosv2r9.aarch64.rpm libmpc-1.1.0-4.eulerosv2r9.aarch64.rpm glibc-devel-2.28-59.eulerosv2r9.aarch64.rpm libxcrypt-devel-4.4.8-4.h2.eulerosv2r9.aarch64.rpm

     rpm -ivh gcc-7.3.0-20190804.h39.eulerosv2r9.aarch64.rpm

     rpm -ivh elfutils-devel-0.179-1.h3.eulerosv2r9.aarch64.rpm

     rpm -ivh make-4.2.1-15.eulerosv2r9.aarch64.rpm

Tlinux 2.4 操作系统

由于内核进行过针对性适配，需要使用版本为8.2.1的GCC编译器，因此编译驱动前需保证8.2.1的GCC编译器已经部署。操作步骤如下：

1. 获取GCC 8.2.1版本编译器的安装包。软件包获取网址：**https://mirrors.tencent.com/tlinux/2.4/arm64/tlinux-sclo/aarch64/tl/devtoolset-8/devtoolset-8-gcc-8.2.1-3.tl2.aarch64.rpm**。
2. 登录运行环境。
3. 上传gcc编译器的安装包到指定路径下，如“/tmp”。
4. 执行以下命令安装GCC编译器。

   rpm -Uvh devtoolset-8-gcc-8.2.1-3.tl2.aarch64.rpm --force --nodeps
5. 安装完成后，执行以下命令查找GCC 8.2.1的可以执行文件所在路径。

   find / -name gcc

   执行文件路径如下红框所示：

   ```
   /opt/rh/devtoolset-8/root/usr/libexec/gcc
   /opt/rh/devtoolset-8/root/usr/bin/gcc
   /opt/rh/devtoolset-8/root/usr/lib/gcc
   ```
6. 配置环境变量。

   export PATH=/opt/rh/devtoolset-8/root/usr/bin/:$PATH
7. 执行以下命令，验证GCC编译器是否安装成功。

   gcc --version

   如出现以下回显，安装后的gcc版本和目标版本一致，则表示安装成功。

   ```
   [root@TENCENT64 bin]# gcc --version
   gcc (GCC) 8.2.1 20180905 (Red Hat 8.2.1-3)
   Copyright (C) 2018 Free Software Foundation, Inc.
   This is free software; see the source for copying conditions.  There is NO
   warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
   ```


版权所有 © 华为技术有限公司

9.5.3.2 安装驱动（适用于.run 格式）

Atlas 300I 推理卡（型号 3000）发布的run安装包是host安装包，用户下载和解压安装包需要在host操作系统完成。本章节介绍宿主机上的安装方法，容器内安装请参见12 容器内安装。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

前提条件

驱动包安装前需要检查并安装相关依赖，具体请参见9.5.3.1 安装依赖。

注意事项

- run安装包支持一键式安装，用户根据安装需求选择对应参数完成安装，具体请参见9.11.1 参数说明。
- 如果有安装NPU 1.X.X系列版本的驱动包，在安装NPU 20.X.X及以上系列版本的驱动包前需要先卸载低版本驱动。
- 使用run安装包后，不要手动设置环境变量export LD_LIBRARY_PATH指向之前驱动包的SO文件，否则可能会出现run安装包内工具连接到之前版本的动态库。指向第三方库文件路径、非run安装包发布库文件路径的配置不受影响。
- 在软件包安装或升级过程中，不允许对Host或Device进行复位或下电操作，否则会出现Device无法正常启动或者升级失败。若出现此问题，解决方法请参见9.10.13 复位或下电导致Device无法正常启动。

安装 NPU 20.X.X 及以上系列版本驱动包

以*A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run*为例介绍.run格式驱动包安装相关指导，具体请根据实际情况进行替换。

须知

首次安装请按照“**驱动->固件**”的顺序；覆盖安装或升级请按照“**固件->驱动**”的顺序，分别安装软件包。固件安装请参见9.6.1 安装固件（适用于.run格式）。

为保证驱动版本、固件版本和MCU版本三者保持配套关系，请参考《**Atlas 300I 推理卡 NPU驱动和固件升级指南（型号 3000, 3010）**》中“升级MCU”章节升级对应的MCU组件。

步骤1 将9.3 获取软件包准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见9.9.5 通过PuTTY登录（网口方式）。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 执行如下命令，进入软件包所在路径（如“/opt”）。

cd */opt*

步骤5 执行如下命令，增加软件包的可执行权限。

chmod +x *A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run*

步骤6 执行如下命令，校验run安装包的一致性和完整性。

./*A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run* **--check**

若出现如下回显信息，表示软件包校验成功。

```
Verifying archive integrity...  100%   SHA256 checksums are OK. All good.
```

说明

软件包校验使用SHA256加密方式；校验过程中若提示：./*A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run* does not contain an embedded MD5 checksum.和./*A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run* does not contain a CRC checksum，表示没有使用MD5和CRC加密方式，可忽略。

步骤7 执行安装命令完成安装。

- 若用户指定安装路径，以“/test/HiAI/”为例。

  可执行**./***A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run* **--full --install-path=** /test/HiAI/命令，完成安装。

- 若未指定安装路径，可执行**./***A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run* **--full**命令，完成安装。

  说明

  - 安装run包驱动时，会将动态库libdcmi.so和头文件dcmi_interface_api.h拷贝到“/usr/local/dcmi/”目录下。
  - run安装包安装过程中出现的异常场景及其解决方法请参见**9.10 常见问题**。
  - 若创建运行用户是非HwHiAiUser，安装驱动软件包时需要指定运行用户（通过**--install-username=***username* **--install-usergroup=***usergroup*参数指定）。
  - 若用户指定安装路径安装驱动时：
    - 若指定路径不存在，则安装时会自动创建目录，若有多层目录，则只有最后一层目录不存在时会自动创建。
    - 若指定路径已存在：
      - 若该路径下所有层级目录属主为root用户，则请确保所有层级目录权限至少为755。
      - 若该路径下有一层目录属主为非root用户，则请用户自行修改为root属主，并确保所有层级目录权限为755。
  - 软件包默认安装路径：/usr/local/Ascend
  - 安装详细日志路径：/var/log/ascend_seclog/ascend_install.log。
  - 安装后软件包的安装路径、安装命令以及运行用户信息记录路径：/etc/ascend_install.info
  - 使用NPU 20.0.X和NPU 20.1.X系列版本驱动包安装时，以非HwHiAiUser用户为运行用户安装驱动，卸载重启后以HwHiAiUser为运行用户再次安装，若出现“/home/HwHiAiUser: Permission denied”报错，请忽略。
  - 使用NPU 20.2.X及以上系列版本软件包安装时，Device侧系统类日志通过msnpureport工具传输到Host侧进行查看。导出操作以及导出日志的存储路径具体请参见**《Ascend 310 黑匣子日志参考》**附录中的“msnpureport工具使用”章节。

    容器内不支持查看Device侧系统类日志，也不支持通过msnpureport工具导出Device侧系统类日志。

步骤8 （可选）对于源码驱动包，安装时若出现以下回显，说明未安装dkms，默认内核源码路径如“/lib/modules/`uname -r`/build”不存在。请根据系统提示输入相应内容：

```
[WARNING]rebuild ko has something wrong, detail in /var/log/ascend_seclog/ascend_rebuild.log
Do you want to try build driver after input kernel absolute path? [y/n]:
```

若需要继续安装，请输入**y**。

出现如下回显后，请根据提示输入您提供的内核源码的实际路径，如：**/lib/modules/`uname -r`/build-bak**。

```
Please input your kernel absolute path:
```

按“Enter”键继续安装。

说明

- 若已安装dkms及相应kernel-header、kernel-devel等组件，则会自动执行dkms驱动编译安装。
- 若未安装dkms，但默认内核源码路径如“/lib/modules/`uname -r`/build”已存在，则会自动使用该内核进行驱动编译。

步骤9 若系统显示安装成功提示信息，则表示驱动已安装成功。根据系统提示信息决定是否重启系统，若需要重启，请执行以下操作；否则，请跳过此步骤。

1. 安装完成后，在host操作系统执行**reboot**命令复位，或将host机下电再上电。
2. host复位后，自动安装驱动及服务，Ascend 310自动进入操作系统。

步骤10 执行**npu-smi info**查看驱动加载是否成功。

若出现如下回显信息，说明加载成功。否则，说明安装失败，请联系华为技术支持处理。

```
+------------------------------------------------------------------------------+
| npu-smi 20.0.1                    Version: 1.73.5.10.050                     |
+-------------------+-----------------+-------------------------------------+
| NPU     Name      | Health          | Power(W)          Temp(C)           |
| Chip    Device    | Bus-Id          | AICore(%)         Memory-Usage(MB)  |
+===================+=================+=====================================+
| 2049    310       | OK              | 12.8              50                |
| 0       0         | 0000:85:00.0    | 0                 2621 / 8192       |
+-------------------+-----------------+-------------------------------------+
| 2049    310       | OK              | 12.8              51                |
| 1       1         | 0000:86:00.0    | 0                 2621 / 8192       |
+-------------------+-----------------+-------------------------------------+
| 2049    310       | OK              | 12.8              52                |
| 2       2         | 0000:87:00.0    | 0                 2621 / 8192       |
+-------------------+-----------------+-------------------------------------+
| 2049    310       | OK              | 12.8              55                |
| 3       3         | 0000:88:00.0    | 0                 2621 / 8192       |
+===================+=================+=====================================+
```

说明

回显中“npu-smi”后面的字段为npu-smi工具版本号；“Version:”后面的字段为驱动版本号。

----结束

后续处理

使用NPU 20.2.X及以上系列版本软件包安装时，请跳过此步骤。使用NPU 20.0.X和NPU 20.1.X系列版本软件包安装时，ADA默认开放侦听端口22118。为确保系统安全，在商用环境中，需要关闭该端口。

须知

1. “/usr/local/Ascend/driver”为使用root用户安装驱动（add-ons随驱动安装）的默认安装路径，下文以此为例来说明操作步骤，请根据实际情况替换。
2. 以root用户安装运行环境时，HwHiAiUser用户为运行环境上的默认进程运行用户，下文以此为例来说明操作步骤。

1. 使用**su - HwHiAiUser**命令切换到运行用户，登录运行环境，打开文件“/home/HwHiAiUser/ide_daemon/ide_daemon.cfg”。
   - 如果SOCK_SWITCH=0，则表示ADA端口已关闭。
   - 如果SOCK_SWITCH=1，则表示ADA端口为开启状态，需要用户手动修改为SOCK_SWITCH=0。

   修改完后执行**wq!**保存退出。
2. 切换到“/usr/local/Ascend/driver/tools”目录下，依次执行如下命令设置环境变量、重启ada进程或执行**reboot**命令重启运行环境。
   a. 设置环境变量。

      export LD_LIBRARY_PATH=/usr/local/Ascend/add-ons:$LD_LIBRARY_PATH
   b. 查找ada进程的ID。

      ps -ef | grep ada
   c. 停止ada进程。

      kill -9 *ada进程ID*
   d. 启动ada进程。

      如果执行如下命令时，提示“ada is exist, don't start again”，则表示系统已自动拉起ada进程，无需再手动执行启动命令。

      /usr/local/Ascend/driver/tools/ada

安装 NPU 1.X.X 系列版本驱动包

以A800-3000-NPU-Ubuntu18.04-ARM-V*x.x.x*.run为例介绍.run格式驱动包安装相关指导，具体请根据实际情况进行替换。

须知

目前仅支持full模式进行安装，即全包安装模式。

步骤1 将**9.3 获取软件包**准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见**9.9.5 通过PuTTY登录（网口方式）**。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 执行如下命令，进入软件包所在路径（如“/opt”）。

cd /*opt*

步骤5 执行如下命令，增加软件包的可执行权限。

chmod +x *A800-3000-NPU-Ubuntu18.04-ARM-Vx.x.x.run*

步骤6 执行如下命令，校验run安装包的一致性和完整性。

./*A800-3000-NPU-Ubuntu18.04-ARM-Vx.x.x.run* **--check**

若出现如下回显信息，表示软件包校验成功。

```
Verifying archive integrity...  100%   SHA256 checksums are OK. All good.
```

步骤7 设置安装路径。

- 如果用户需要指定安装路径，则需要先创建安装路径。以“/test/HiAI/”为例，用户先执行**mkdir /test/HiAI/**命令，完成安装路径的创建，再选择该安装路径进行软件安装。
- 如果用户不需要指定安装路径，安装包会安装到默认路径“/usr/local/HiAI”下。

步骤8 执行安装命令完成安装。

安装过程支持一键式安装操作。

- 若用户指定安装路径，以“/test/HiAI/”为例。

  可执行**./***A800-3000-NPU-Ubuntu18.04-ARM-Vx.x.x.run* **--full --path /test/HiAI/**命令，完成安装。
- 若未指定安装路径，可执行**./***A800-3000-NPU-Ubuntu18.04-ARM-Vx.x.x.run* **--full**命令，完成安装。

说明

- 安装run包驱动时，会将动态库libdcmi.so和头文件dcmi_interface_api.h拷贝到“/usr/local/dcmi/”目录下。
- run安装包安装过程中出现的异常场景及其解决方法请参见**9.10 常见问题**。
- 安装过程中Ascend 310驱动、固件安装和升级的日志信息输出在“/var/log/HiAI-install.log”文件中，用户可执行**./****.run* **--viewlog**命令，打开日志；npu-smi工具安装的日志信息输出在“/var/log/it_logs”中。
- 安装驱动过程中出现“drv_dkms_env_check failed, details in : /var/log/HiAI-install.log”的报错，请查看日志信息，日志中显示“The program 'dkms' is currently not installed.”，说明错误为缺少DKMS，请安装DKMS。具体操作请参见**9.10.3 安装驱动过程中出现“drv_dkms_env_check failed, details in : /var/log/HiAI-install.log”的报错**。

步骤9 run安装包安装完成后需要复位host主机，复位后，运行的Ascend 310软件和mini固件即为安装的最新版本。复位host主机步骤如下。

1. 安装完成后，在host操作系统执行**reboot**命令复位，或将host机下电再上电。
2. host复位后，自动安装驱动及服务，Ascend 310自动进入操作系统。

步骤10 执行**npu-smi info**查看驱动加载是否成功。

若出现如下回显信息，说明加载成功。否则，说明加载失败，请联系华为技术支持处理。

```
root@atlas-G560:~# npu-smi info
+------------------------------------------------------------------------------+
| npu-smi 1.1.2                    Version: 1.1.T19.B880                       |
+-------------------+-----------------+-------------------------------------+
| NPU     Name      | Health          | Power(W)          Temp(C)           |
| Chip              | Bus-Id          | AICore(%)         Memory-Usage(MB)  |
+===================+=================+=====================================+
| 77      HI1910    | Warning         | 12.8              61                |
| 0                 | 0000:4F:00.0    | 0                 2457 / 8192       |
```

```
+-------------------+-----------------+----------------------------------------+
| 77      HI1910    | OK              | 12.8            58                     |
| 1                 | 0000:53:00.0    | 0               2375 / 8192            |
+-------------------+-----------------+----------------------------------------+
| 77      HI1910    | OK              | 12.8            57                     |
| 2                 | 0000:57:00.0    | 0               2375 / 8192            |
+-------------------+-----------------+----------------------------------------+
| 77      HI1910    | OK              | 12.8            58                     |
| 3                 | 0000:5B:00.0    | 0               2375 / 8192            |
+===================+=================+========================================+
```

说明

回显中“npu-smi”后面的字段为npu-smi工具版本号，Version:”后面的字段为驱动/芯片固件版本号。

----结束

9.5.4 重构驱动包安装

本章内容仅适用于NPU 20.1.X及以上系列版本。

9.5.4.1 安装依赖

GCC 和内核源码

确认目标OS及其内核版本号对应的GCC及内核源码是否已存在，具体请参见**9.5.3.1 安装依赖**。

makeself 2.4.0 工具

下载开源工具：makeself 2.4.0。下载地址请参见：**https://github.com/megastep/makeself**或**https://github.com/megastep/makeself/releases/tag/release-2.4.0**。

pigz 工具

执行如下命令，检查pigz工具是否已安装。例如：

- Ubuntu操作系统：

  dpkg-query -s pigz

- CentOS操作系统：

  rpm -qa |grep pigz

若未安装请使用对应的软件管理器进行安装。例如：

- Ubuntu操作系统：

  apt install pigz

- CentOS操作系统：

  yum install pigz

9.5.4.2 重构驱动包

用户可以根据本章节内容，构建一个满足目标OS版本，内核版本，gcc版本要求的驱动安装包。

版权所有 © 华为技术有限公司

前提条件

编译前需要安装相关依赖，具体请参见9.5.4.1 安装依赖。

操作步骤

步骤1 根据9.3 获取软件包获取源码驱动包和重构工具包。

步骤2 上传源码驱动包、重构工具包、开源软件“makeself-release-2.4.0.zip”到运行环境的同一目录。如：/opt下。

步骤3 执行以下命令，进入步骤2软件包所在路径，如/opt下。

cd /*opt*

步骤4 执行以下命令，增加源码驱动包的执行权限。

命令：**chmod +x** **.run*

示例：**chmod +x A300-3000-npu-driver_20.1.0_linux-aarch64.run**

步骤5 执行以下命令，解压源码驱动包到指定目录。

命令：**./****.run* **--noexec --extract=***目标路径*

示例：**./A300-3000-npu-driver_20.1.0_linux-aarch64.run --noexec --extract=./tmp**

步骤6 执行以下命令，解压重构工具包到当前目录。

命令：**tar -xzvf** *重构工具包名称* **--strip-components 1**

示例：**tar -xzvf A300-3000-npu-driver-repack-tools_20.1.0.tar.gz --strip-components 1**

步骤7 执行以下命令，解压软件包“makeself-release-2.4.0.zip”。

unzip makeself-release-2.4.0.zip

步骤8 执行以下命令，构建新驱动包。

命令：**bash build.sh** *makeself.sh路径 包解压的路径 输出包名*

示例：**bash build.sh ./makeself-release-2.4.0/makeself.sh tmp/ A300-3000-npu-driver-repack.run**

步骤9 请参见9.5.4.3 安装驱动（适用于.run格式），安装驱动。

----结束

9.5.4.3 安装驱动（适用于.run 格式）

Atlas 300I 推理卡（型号 3000）发布的run安装包是host安装包，用户下载和解压安装包需要在host操作系统完成。本章节介绍宿主机上的安装方法，容器内安装请参见12 容器内安装。

本章以重构驱动包A300-3000-npu-driver-repack.run为例介绍.run格式驱动包安装相关指导，具体请根据实际情况进行替换。

注意事项

- run安装包支持一键式安装，用户根据安装需求选择对应参数完成安装，具体请参见**9.11.1 参数说明**。
- 如果有安装NPU 1.X.X系列版本的驱动包，在安装NPU 20.X.X及以上系列版本的驱动包前需要先卸载低版本驱动。
- 使用run安装包后，不要手动设置环境变量export LD_LIBRARY_PATH指向之前驱动包的SO文件，否则可能会出现run安装包内工具连接到之前版本的动态库。指向第三方库文件路径、非run安装包发布库文件路径的配置不受影响。
- 在软件包安装或升级过程中，不允许对Host或Device进行复位或下电操作，否则会出现Device无法正常启动或者升级失败。若出现此问题，解决方法请参见**9.10.13 复位或下电导致Device无法正常启动**。

操作步骤

须知

首次安装请按照“**驱动->固件**”的顺序；覆盖安装或升级请按照“**固件->驱动**”的顺序，分别安装软件包。固件安装请参见**9.6.1 安装固件（适用于.run格式）**。

为保证驱动版本、固件版本和MCU版本三者保持配套关系，请参考《**Atlas 300I 推理卡 NPU驱动和固件升级指南（型号 3000, 3010）**》中“升级MCU”章节升级对应的MCU组件。

步骤1 将**9.5.4.2 重构驱动包**构建的驱动包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见**9.9.5 通过PuTTY登录（网口方式）**。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 执行如下命令，进入软件包所在路径（如/opt）。

cd */opt*

步骤5 执行如下命令，校验run安装包的一致性和完整性。

./*A300-3000-npu-driver-repack.run* **--check**

出现如下回显信息，表示软件包校验成功。

```
Verifying archive integrity...  100%   SHA256 checksums are OK. All good.
```

说明

软件包校验使用SHA256加密方式；校验过程中若提示：./*A300-3000-npu-driver-repack.run* does not contain an embedded MD5 checksum.和./*A300-3000-npu-driver-repack.run* does not contain a CRC checksum，表示没有使用MD5和CRC加密方式，可忽略。

步骤6 执行安装命令完成安装。

- 若用户指定安装路径，以“/test/HiAI/”为例。

  可执行**./***A300-3000-npu-driver-repack.run* **--full --install-path=** /test/HiAI/命令，完成安装。

- 若未指定安装路径，可执行**./***A300-3000-npu-driver-repack.run* **--full**命令，完成安装。

  说明

  - 安装run包驱动时，会将动态库libdcmi.so和头文件dcmi_interface_api.h拷贝到“/usr/local/dcmi/”目录下。
  - run安装包安装过程中出现的异常场景及其解决方法请参见**9.10 常见问题**。
  - 若创建运行用户是非HwHiAiUser，安装驱动软件包时需要指定运行用户（通过**--install-username=***username* **--install-usergroup=***usergroup*参数指定）。
  - 若用户指定安装路径安装驱动时：
    - 若指定路径不存在，则安装时会自动创建目录，若有多层目录，则只有最后一层目录不存在时会自动创建。
    - 若指定路径已存在：
      - 若该路径下所有层级目录属主为root用户，则请确保所有层级目录权限至少为755。
      - 若该路径下有一层目录属主为非root用户，则请用户自行修改为root属主，并确保所有层级目录权限为755。
  - 软件包默认安装路径：/usr/local/Ascend
  - 安装详细日志路径：/var/log/ascend_seclog/ascend_install.log。
  - 安装后软件包的安装路径、安装命令以及运行用户信息记录路径：/etc/ascend_install.info
  - 使用NPU 20.0.X和NPU 20.1.X系列版本驱动包安装时，以非HwHiAiUser用户为运行用户安装驱动，卸载重启后以HwHiAiUser为运行用户再次安装，若出现“/home/HwHiAiUser: Permission denied”报错，请忽略。
  - 使用NPU 20.2.X及以上系列版本软件包安装时，Device侧系统类日志通过msnpureport工具传输到Host侧进行查看。导出操作以及导出日志的存储路径具体请参见《**Ascend 310 黑匣子日志参考**》附录中的“msnpureport工具使用”章节。

    容器内不支持查看Device侧系统类日志，也不支持通过msnpureport工具导出Device侧系统类日志。

步骤7 若系统显示安装成功提示信息，则表示驱动已安装成功。根据系统提示信息决定是否重启系统，若需要重启，请执行以下操作；否则，请跳过此步骤。

1. 安装完成后，在host操作系统执行**reboot**命令复位，或将host机下电再上电。
2. host复位后，自动安装驱动及服务，Ascend 310自动进入操作系统。

步骤8 执行**npu-smi info**查看驱动加载是否成功。

回显如下类似信息，说明加载成功。否则，说明安装失败，请联系华为技术支持处理。

```
+------------------------------------------------------------------------------+
| npu-smi 20.0.1                    Version: 1.73.5.10.050                     |
+-------------------+-----------------+-------------------------------------+
| NPU     Name      | Health          | Power(W)          Temp(C)           |
| Chip    Device    | Bus-Id          | AICore(%)         Memory-Usage(MB)  |
+===================+=================+=====================================+
| 2049    310       | OK              | 12.8              50                |
| 0       0         | 0000:85:00.0    | 0                 2621 / 8192       |
+-------------------+-----------------+-------------------------------------+
| 2049    310       | OK              | 12.8              51                |
| 1       1         | 0000:86:00.0    | 0                 2621 / 8192       |
+-------------------+-----------------+-------------------------------------+
| 2049    310       | OK              | 12.8              52                |
| 2       2         | 0000:87:00.0    | 0                 2621 / 8192       |
+-------------------+-----------------+-------------------------------------+
| 2049    310       | OK              | 12.8              55                |
```


版权所有 © 华为技术有限公司

```
| 3       3        | 0000:88:00.0    | 0                2621 / 8192         |
+===================+=================+=======================================+
```

说明

回显中“npu-smi”后面的字段为npu-smi工具版本号；“Version:”后面的字段为驱动版本号。

----结束

后续处理

使用NPU 20.2.X及以上系列版本软件包安装时，请跳过此步骤。使用NPU 20.0.X和NPU 20.1.X系列版本软件包安装时，ADA默认开放侦听端口22118。为确保系统安全，在商用环境中，需要关闭该端口。

须知

1. “/usr/local/Ascend/driver”为使用root用户安装驱动（add-ons随驱动安装）的默认安装路径，下文以此为例来说明操作步骤，请根据实际情况替换。
2. 以root用户安装运行环境时，HwHiAiUser用户为运行环境上的默认进程运行用户，下文以此为例来说明操作步骤。

1. 使用**su - HwHiAiUser**命令切换到运行用户，登录运行环境，打开文件“/home/HwHiAiUser/ide_daemon/ide_daemon.cfg”。
   - 如果SOCK_SWITCH=0，则表示ADA端口已关闭。
   - 如果SOCK_SWITCH=1，则表示ADA端口为开启状态，需要用户手动修改为SOCK_SWITCH=0。

   修改完后执行**wq!**保存退出。
2. 切换到“/usr/local/Ascend/driver/tools”目录下，依次执行如下命令设置环境变量、重启ada进程或执行**reboot**命令重启运行环境。
   a. 设置环境变量。

      export LD_LIBRARY_PATH=/usr/local/Ascend/add-ons:$LD_LIBRARY_PATH

   b. 查找ada进程的ID。

      ps -ef | grep ada

   c. 停止ada进程。

      kill -9 *ada进程ID*

   d. 启动ada进程。

      如果执行如下命令时，提示“ada is exist, don't start again”，则表示系统已自动拉起ada进程，无需再手动执行启动命令。

      /usr/local/Ascend/driver/tools/ada

9.6 安装固件

本章节主要介绍固件在宿主机上的安装方法，不支持在docker容器内安装。

版权所有 © 华为技术有限公司

9.6.1 安装固件（适用于.run 格式）

使用NPU 1.X.X系列版本软件包安装时，固件包集成在驱动包内，安装或升级驱动过程中会自动安装和升级固件。具体安装方式请参见**9.5 安装驱动**。

使用NPU 20.X.X及以上系列版本软件包安装时，首次安装请按照“**驱动->固件**”的顺序；覆盖安装或升级请按照“**固件->驱动**”的顺序，分别安装软件包，驱动安装请参见**9.5 安装驱动**，本章节以A300-3000-3010-npu-firmware_*<version>*.run为例介绍宿主机上.run格式固件包的安装方法。

注意事项

- run安装包支持一键式安装，用户根据安装需求选择对应参数完成安装，具体请参见**9.11.1 参数说明**。
- 使用run安装包后，不要手动设置环境变量export LD_LIBRARY_PATH指向之前驱动包的SO文件，否则可能会出现run安装包内工具连接到之前版本的动态库。指向第三方库文件路径、非run安装包发布库文件路径的配置不受影响。
- 在软件包安装或升级过程中，不允许对Host或Device进行复位或下电操作，否则会出现Device无法正常启动或者升级失败。若出现此问题，解决方法请参见**9.10.13 复位或下电导致Device无法正常启动**。
- 扩容或备件替换场景下，请执行**npu-smi info -t board -i** *id* 命令查询相应的版本号，建议服务器上所有Atlas 300I 推理卡的固件版本保持一致，若不一致，请先进行固件升级，否则可能导致服务器无法识别Atlas 300I 推理卡或Atlas 300I 推理卡在业务运行过程中出现丢卡等问题。具体请参见**9.10.15 固件版本不一致导致服务器无法识别Atlas 300I 推理卡**。

操作步骤

步骤1 将**9.3 获取软件包**准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见**9.9.5 通过PuTTY登录（网口方式）**。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 执行如下命令，进入软件包所在路径（如“/opt”）。

cd */opt*

步骤5 执行如下命令，增加软件包的可执行权限。

chmod +x *A300-3000-3010-npu-firmware_<version>.run*

步骤6 执行如下命令，校验run安装包的一致性和完整性。

./*A300-3000-3010-npu-firmware_<version>.run* **--check**

出现如下回显信息，表示软件包校验成功。

```
Verifying archive integrity...  100%   SHA256 checksums are OK. All good.
```

说明

软件包校验使用SHA256加密方式；校验过程中若提示：*./A300-3000-3010-npu-firmware_<version>.run* does not contain an embedded MD5 checksum.和./*A300-3000-3010-npu-firmware_<version>.run* does not contain a CRC checksum，表示没有使用MD5和CRC加密方式，可忽略。

版权所有 © 华为技术有限公司

步骤7 执行如下安装命令完成安装。

./*A300-3000-3010-npu-firmware_<version>.run* **--full**

说明

- 安装过程中出现的异常场景及其解决方法请参见**9.10 常见问题**。
- 安装固件软件包不支持指定运行用户名和用户组，共用驱动的运行用户名和用户组。
- 安装固件软件包不支持指定路径，共用驱动的安装路径。
- 软件包默认安装路径：/usr/local/Ascend
- 安装详细日志路径：/var/log/ascend_seclog/ascend_install.log。
- 安装后软件包的安装路径、安装命令以及运行用户信息记录路径：/etc/ascend_install.info
- 使用NPU 20.2.X及以上系列版本软件包安装时，Device侧系统类日志通过msnpureport工具传输到Host侧进行查看。导出操作以及导出日志的存储路径具体请参见**《Ascend 310 黑匣子日志参考》**附录中的“msnpureport工具使用”章节。

  容器内不支持查看Device侧系统类日志，也不支持通过msnpureport工具导出Device侧系统类日志。

步骤8 若系统显示安装成功提示信息，则表示固件已安装成功。根据系统提示信息决定是否重启系统，若需要重启，请执行以下操作；否则，请跳过此步骤。

1. 安装完成后，在host操作系统执行**reboot**命令复位，或将host机下电再上电。
2. host复位后，Ascend 310自动进入操作系统。

步骤9 执行如下命令查看芯片固件版本号。若与目标版本一致，则说明安装成功。

*/usr/local/Ascend/***driver/tools/upgrade-tool --device_index -1 --component -1 --version**

说明

如果安装驱动时指定安装路径，则命令中的“*/usr/local/Ascend*”请根据实际情况替换。

----结束

9.6.2 安装固件（适用于.rpm 格式）

使用NPU 1.X.X系列版本软件包安装时，固件包集成在驱动包内，安装或升级驱动过程中会自动安装和升级固件。具体安装方式请参见**9.5.2.2 安装驱动（适用于.rpm格式）**。

使用NPU 20.X.X及以上系列版本软件包安装时，请按照“**驱动->固件**”的顺序，分别安装软件包，驱动安装请参见**9.5.2.2 安装驱动（适用于.rpm格式）**。本章节以A300-3000-npu-firmware_*<version>*.rpm为例介绍宿主机上.rpm格式固件包的安装方法，具体请根据实际情况进行替换。

注意事项

- 当前rpm包仅适配EulerOS系统，其他环境请不要使用。
- 当前rpm包安装不支持修改安装路径功能，请按照正常rpm流程安装，不支持rpm包的--force命令安装升级。
- rpm包与run包为同级别组件，两者不能兼容，即安装run包后不能再安装rpm包、安装rpm包后也不支持再安装run包。
- Ascend 310内部部分功能，只支持HwHiAiUser用户访问和执行，用户首次安装时系统会自动创建HwHiAiUser用户。

版权所有 © 华为技术有限公司

- 安装完成后需要重启生效。卸载完成后，如需再次安装，需要先重启再安装。
- 如果已安装NPU 1.X.X系列版本的rpm包，在安装NPU 20.X.X及以上系列版本的rpm包前需要先卸载原版本再安装。
- 在软件包安装或升级过程中，不允许对Host或Device进行复位或下电操作，否则会出现Device无法正常启动或者升级失败。若出现此问题，解决方法请参见**9.10.13 复位或下电导致Device无法正常启动**。
- 扩容或备件替换场景下，请执行**npu-smi info -t board -i** *id* 命令查询相应的版本号，建议服务器上所有Atlas 300I 推理卡的固件版本保持一致，若不一致，请先进行固件升级，否则可能导致服务器无法识别Atlas 300I 推理卡或Atlas 300I 推理卡在业务运行过程中出现丢卡等问题。具体请参见**9.10.15 固件版本不一致导致服务器无法识别Atlas 300I 推理卡**。

操作步骤

步骤1 将**9.3 获取软件包**准备的安装包上传至Linux系统任意目录下（如“/opt”）。

步骤2 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见**9.9.5 通过PuTTY登录（网口方式）**。

步骤3 执行如下命令，切换至root用户。

su - root

步骤4 在软件包所在路径执行如下命令完成Host侧固件包安装。

rpm -ivh *A300-3000-npu-firmware_<version>.rpm*

说明

软件包安装过程中，若无HwHiAiUser用户系统会自动创建该用户。

步骤5 执行**reboot**命令重启Host侧系统使固件版本生效。

步骤6 执行**d-agent -v**命令查看固件安装是否成功。

回显如下类似信息，说明安装成功。否则，说明安装失败，请联系华为工程师处理。

```
driver: 20.1.0
firmware: 1.76.T20.0.B200
```

说明

安装过程中的日志信息输出在“/var/log/ascend_seclog/ascend_install.log”文件中，用户可以执行**vim /var/log/ascend_seclog/ascend_install.log**命令打开日志。

步骤7 固件安装成功后，用户需要升级Atlas 300I 推理卡（型号 3000）的芯片固件。

- 执行**d-agent -b 0000:***91:00.0***命令完成单个芯片的固件升级。
- Atlas 300I 推理卡（型号 3000）所有芯片上无业务进程的情况下，执行**d-agent -a**命令完成所有芯片固件升级。

须知

- Atlas 300I 推理卡（型号 3000）芯片的ID可通过**lspci | grep d100**命令查看。
- 单个芯片固件升级大概需要30s；所有芯片固件升级持续时间具体以实际为准。需等待固件升级完成后，再查询固件版本。

执行**d-agent -g 0000:***91:00.0*命令查看升级后固件版本号。回显如下类似信息，如果版本号与目标版本一致，说明升级成功。否则，说明升级失败，请联系华为技术工程师处理。nve:1.76.20.0.B200 xloader:1.76.20.0.B200

----结束

9.7 卸载驱动

9.7.1 卸载驱动（适用于.run 格式）

目前发布的run包在宿主机上的卸载方式相同。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

前提条件

未安装run安装包时不能使用卸载功能。

卸载 NPU 20.X.X 及以上系列版本驱动

以A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run为例介绍.run格式驱动包卸载相关操作，具体请根据实际情况进行替换。

步骤1 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见**9.9.5 通过PuTTY登录（网口方式）**。

步骤2 执行如下命令，切换至root用户。

su - root

步骤3 支持以下两种卸载方式，用户根据实际情况选择其中一个方式卸载即可：

- 在任意路径执行如下命令卸载软件包：

  <install-path>/**driver/script/uninstall.sh**

  说明

  *<install-path>*表示软件包安装路径，可以执行**cat /etc/ascend_install.info**命令查询安装路径；请根据实际情况替换。

- 在软件包所在路径执行以下命令进行卸载：

  ./*A300-3000-3010-npu-driver_x.x.x_ubuntu18.04-aarch64.run* **--uninstall**

如果卸载过程中无错误信息提示，则表示卸载成功，根据系统提示信息决定是否重启服务器，完成对软件包的卸载。

步骤4 若需要重启系统，请执行以下命令；否则，请跳过此步骤。

reboot

须知

驱动卸载后，npu-smi工具不可用。

----结束

卸载 NPU 1.X.X 系列版本驱动

以A800-3000-NPU-Ubuntu18.04-ARM-V*x.x.x*.run为例介绍.run格式驱动包卸载相关操作，具体请根据实际情况进行替换。

步骤1 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见**9.9.5 通过PuTTY登录（网口方式）**。

步骤2 执行如下命令，切换至root用户。

su - root

步骤3 执行如下命令，进入软件包所在路径（如“/opt”）。

cd */opt*

步骤4 执行如下命令卸载驱动。

./*A800-3000-NPU-Ubuntu18.04-ARM-Vx.x.x.run* **--uninstall**

如果卸载过程中无错误信息提示，则表示卸载成功。请继续执行**步骤5**。

步骤5 若需要重启系统，请执行以下命令；否则，请跳过此步骤。

reboot

须知

驱动卸载后，npu-smi工具不可用。

----结束

9.7.2 卸载驱动（适用于.rpm 格式）

本章主要介绍Host侧驱动卸载的相关指导。

前提条件

未安装rpm安装包时不能使用卸载功能。

卸载 NPU 20.X.X 及以上系列版本驱动

须知

npu-smi工具集成在驱动包内，卸载驱动过程中会自动卸载npu-smi工具。固件包支持单独卸载，驱动和固件卸载没有先后顺序要求，固件卸载请参见**9.8.2 卸载固件（适用于.rpm格式）**。

版权所有 © 华为技术有限公司

步骤1 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见**9.9.5 通过PuTTY登录（网口方式）**。

步骤2 执行如下命令，切换至root用户。

su - root

步骤3 执行如下命令查看rpm驱动列表。

rpm -qa | grep npu

回显如下，表示存在如下驱动：

```
npu-driver-20.2.0-1.aarch64
```

步骤4 执行如下命令卸载驱动。

rpm -e *npu-driver-20.2.0-1.aarch64*

说明

上述命令中软件名称请根据**步骤3**的查询情况进行替换。

步骤5 （可选）卸载完成后，如果需要再次安装软件包，请复位host主机。

reboot

----结束

卸载 NPU 1.X.X 系列版本驱动

须知

固件包集成在驱动包内，卸载驱动过程中会自动卸载固件。npu-smi工具支持单独卸载，具体请参见**10.1.5 卸载npu-smi工具**。

步骤1 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见**9.9.5 通过PuTTY登录（网口方式）**。

步骤2 执行如下命令，切换至root用户。

su - root

步骤3 执行如下命令查看rpm驱动列表。

rpm -qa | grep HiAI

回显如下，表示存在如下驱动：

```
HiAI_Euler_V2R8_driver-100.1.32.080-1.aarch64
```

步骤4 执行如下命令卸载驱动。

rpm -e *HiAI_Euler_V2R8_driver-100.1.32.080-1.aarch64*

说明

上述命令中软件名称请根据**步骤3**的查询情况进行替换。

步骤5 （可选）卸载完成后，如果需要再次安装软件包，请复位host主机。

reboot

----结束

9.8 卸载固件

9.8.1 卸载固件（适用于.run 格式）

卸载NPU 1.X.X系列版本软件包时，固件包集成在驱动包内，卸载驱动会自动卸载固件。具体请参见**9.7.1 卸载驱动（适用于.run格式）**。

卸载NPU 20.X.X及以上系列版本软件包时，需要分别单独卸载驱动和固件，驱动和固件的卸载没有先后顺序要求。驱动卸载请参见**9.7.1 卸载驱动（适用于.run格式）**。本章以A300-3000-3010-npu-firmware_*<version>*.run为例介绍固件卸载相关指导，具体请根据实际情况进行替换。

前提条件

未安装run安装包时不能使用卸载功能。

操作步骤

步骤1 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见**9.9.5 通过PuTTY登录（网口方式）**。

步骤2 执行如下命令，切换至root用户。

su - root

步骤3 支持以下两种方式，用户根据实际情况选择其中一个方式卸载即可：

- 在任意路径执行如下命令卸载软件包：

  <install-path>/**firmware/script/uninstall.sh**

  说明

  *<install-path>*表示软件包安装路径，可以执行**cat /etc/ascend_install.info**命令查询安装路径；请根据实际情况替换。

- 在软件包所在路径执行以下命令进行卸载：

  ./*A300-3000-3010-npu-firmware_<version>.run* **--uninstall**

如果卸载过程中无错误信息提示，则表示卸载成功，根据系统提示信息决定是否重启服务器，完成对软件包的卸载。

步骤4 若需要重启系统，请执行以下命令；否则，请跳过此步骤。

reboot

----结束

9.8.2 卸载固件（适用于.rpm 格式）

卸载NPU 1.X.X系列版本软件包时，固件包集成在驱动包内，卸载驱动会自动卸载固件。具体请参见**9.7.2 卸载驱动（适用于.rpm格式）**。

卸载NPU 20.X.X及以上系列版本软件包时，需要分别单独卸载驱动和固件，驱动和固件的卸载没有先后顺序要求。驱动卸载请参见**9.7.2 卸载驱动（适用于.rpm格式）**。本章主要介绍Host侧固件卸载的相关指导。

前提条件

未安装run安装包时不能使用卸载功能。

操作步骤

步骤1 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见**9.9.5 通过PuTTY登录（网口方式）**。

步骤2 执行如下命令，切换至root用户。

su - root

步骤3 执行如下命令查看rpm固件列表。

rpm -qa | grep npu

回显如下，表示存在如下固件：

```
npu-firmware-20.2.0-1.noarch
```

步骤4 执行如下命令卸载固件。

rpm -e *npu-firmware-20.2.0-1.noarch*

说明

上述命令中软件名称请根据**步骤3**的查询情况进行替换。

步骤5 （可选）卸载完成后，如果需要再次安装软件包，请复位host主机。

reboot

----结束

9.9 常用操作

9.9.1 使用 WinSCP 传输文件

操作场景

在PC机上使用WinSCP工具进行文件传输。

必备事项

前提条件

目的设备已开启SFTP服务。

数据

需准备如下数据：

- 待连接服务器的IP地址
- 登录待连接服务器的用户名和密码

软件

WinSCP.exe：此工具为第三方软件，请自行准备。

操作步骤

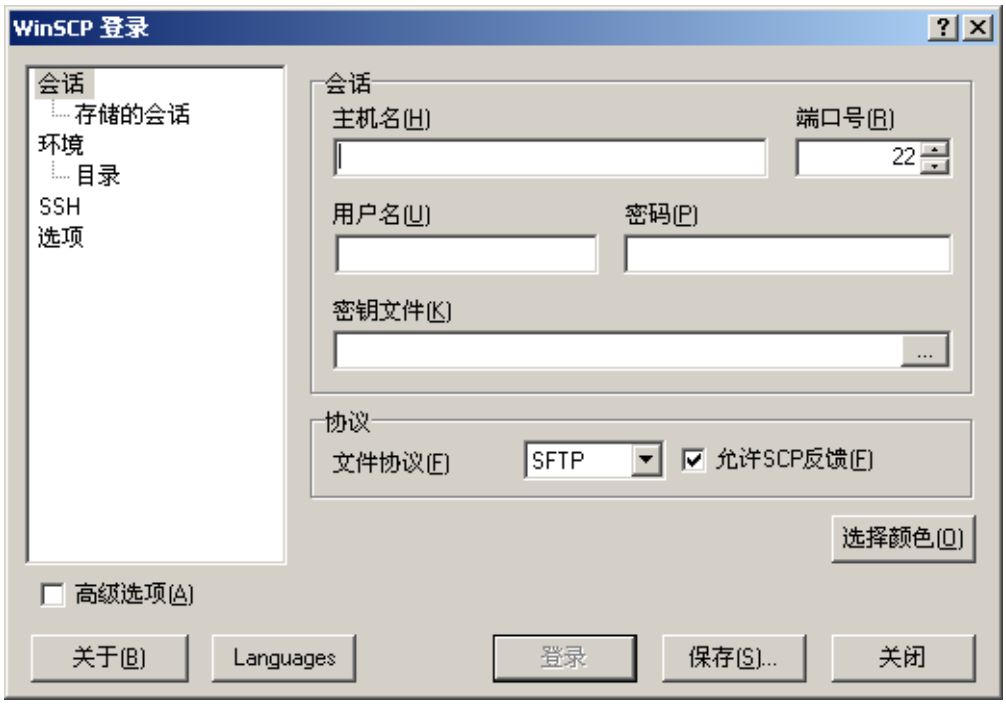
步骤1 打开“WinSCP”文件夹，双击“WinSCP.exe”。

弹出“WinSCP 登录”对话框，如图9-2所示。

说明

若系统非中文操作系统，可以单击“Languages”进行界面语言的选择。

图 9-2 WinSCP 登录



步骤2 单击“会话”。

步骤3 设置登录参数。

参数说明如下。

- 主机名（H）：待连接设备的IP地址，例如“192.168.2.10”。
- 端口号（R）：默认为“22”。
- 用户名（U）：待连接设备的操作系统用户名，例如“admin123”。
- 密码（P）：待连接设备的操作系统用户的密码，例如“admin123”。
- 密钥文件（K）：默认为空，保留默认值。
- 协议：选择默认文件协议“SFTP”，并勾选“允许SCP反馈（F）”。

步骤4 单击“登录”。

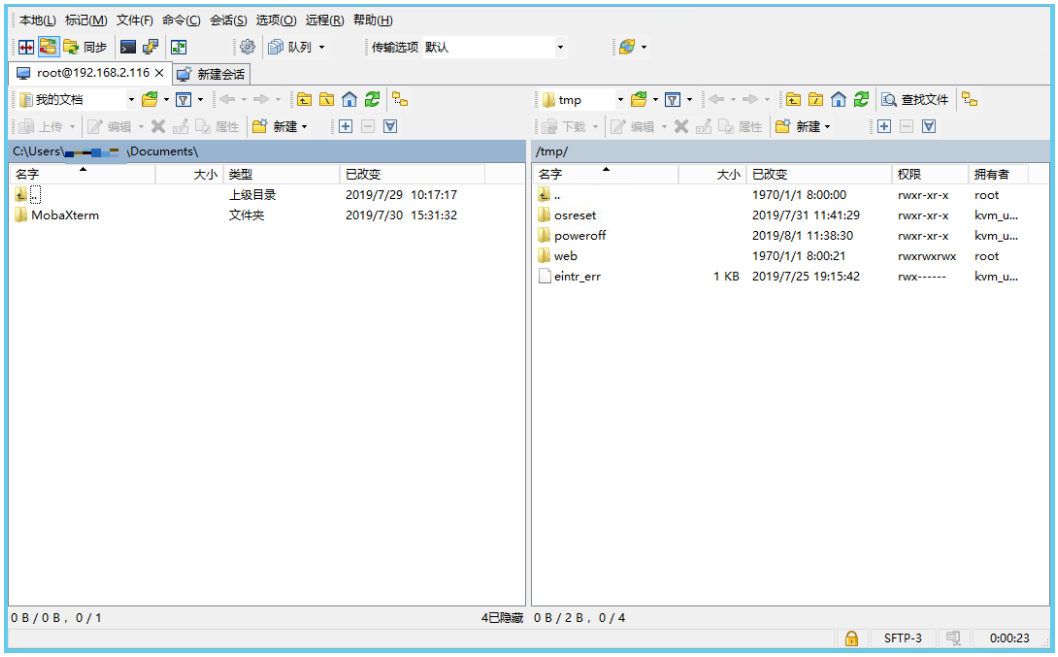
进入“WinSCP”文件传输界面。

说明

如果首次登录时没有选择密钥文件，此时会弹出一个警告提示框，询问“是否连接并添加密钥到缓存？”，单击“是（Y）”，进入“WinSCP”文件传输界面。

步骤5 根据实际需求，在界面左右区的指定目录中进行文件夹的创建、删除和复制等操作。

图 9-3 WinSCP 界面



说明

界面左侧区域为本地PC的目录，右侧区域为已连接设备的目录。

----结束

9.9.2 设置用户有效期

为保证用户的安全性，应设置用户的有效期，使用系统命令**chage**来设置用户的有效期。

命令为：

chage [**-m** *mindays*] [**-M** *maxdays*] [**-d** *lastday*] [**-I** *inactive*] [**-E** *expiredate*] [**-W** *warndays*] *user*

相关参数请参见**表9-5**。

版权所有 © 华为技术有限公司

表 9-5 设置用户有效期

参数	参数说明
-m	密码可更改的最小天数。设置为“0”表示任何时候都可以更改密码。
-M	密码保持有效的最大天数。设置为“-1”表示可删除这项密码的检测，用户密码则不会超期，存在安全风险，请谨慎使用。
-d	上一次更改密码的日期。
-I	停滞时期。过期指定天数后，设定密码为失效状态。
-E	用户到期的日期。超过该日期，此用户将不可用。
-W	密码到期前，提前收到警告信息的天数。
-l	列出用户以及密码的有效期。由非特权用户来确定用户或密码何时过期。

说明

- 表9-5只列举出常用的参数，用户可通过**chage --help**命令查询详细的参数说明。
- 日期格式为YYYY-MM-DD，如**chage -E** 2019-12-01 *test*表示用户*test*的用户在2019年12月1日过期。
- User如果不填写，默认为root用户。

例如：修改用户*test*的有效期为2019年12月31日。可执行如下命令：

chage -E *2019-12-31 test*

9.9.3 配置 Host 侧通过 SSH 方式登录 Device 的 IP

前提条件

本章节内容仅适用于NPU 20.X.X及以上系列版本。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

操作步骤

步骤1 使用**npu-smi set -t ssh-enable**命令打开SSH开关，命令的具体使用方法请参见**10.1.4.6.1 配置所有芯片的ssh使能状态**和**10.1.4.6.2 配置指定芯片的ssh使能状态**。

步骤2 在Host侧配置IP。IP必须与Device侧的IP为同一网段，如：192.168.1.100。Device默认IP地址从192.168.1.199依次递减，例如用户有多个Device，则IP地址分别为192.168.1.199，192.168.1.198，192.168.1.197，...，最多支持配置64个Device。本文仅给出Host操作系统为Ubuntu 18.04和CentOS 7.6配置IP的方法，其他OS配置IP的方法可参考对应OS官网社区的FAQ去解决。

版权所有 © 华为技术有限公司

- 在Host操作系统为Ubuntu 18.04配置静态IP
  a. 修改“/etc/netplan”目录下的yaml配置文件。
     i. 执行以下命令，进入“/etc/netplan”目录下查看配置文件名称。
     cd /etc/netplan
     ls
     查询结果如下：
     ```
     01-netcfg.yaml
     ```
     ii. 执行以下命令，打开配置文件。
     vi /etc/netplan/*01-netcfg.yaml*
     在该文件中增加以下内容：
     ```
     endvnic:
             addresses:   [192.168.1.100/24]
             nameservers:
                 addresses:    [255.255.254.0]
     ```
     按“Esc”切换到命令模式，输入**:wq**保存并退出。
  b. 执行以下命令，使网络配置生效。
     netplan apply
  c. 执行**ifconfig**命令，查看网络配置是否成功。
     若出现如下回显，表示网络配置成功。否则，请联系华为技术支持处理。
     ```
     endvnic: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
         inet 192.168.1.100  netmask 255.255.255.0  broadcast 192.168.1.255
         inet6 fe80::xxxx:xxxx:xxxx:74d3  prefixlen 64  scopeid 0x20<link>
         ...
     ```
- 在Host操作系统为CentOS 7.6配置动态IP（动态添加IP，每次重启后都要重新操作）
  a. **systemctl status NetworkManager**
     如果不是active的，需要启动**systemctl start NetworkManager**
  b. **ifconfig -s endvnic 192.168.1.100**
  c. **systemctl disable NetworkManager**
  d. **systemctl stop NetworkManager**
- 在Host操作系统为CentOS 7.6配置静态IP（静态设置，每次重启无需重复操作）
  a. **systemctl status NetworkManager**
     如果不是active的，需要通过如下命令配置开机自启动。
     systemctl start NetworkManager
     systemctl enable NetworkManager
  b. **nmcli connect add autoconnect no type ethernet con-name endvnic ifname endvnic** 生成“/etc/sysconfig/network-scripts/ifcfg-endvnic”。
  c. 使用**ip a** 查看MAC地址，例如：
     ```
     3: endvnic: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
     link/ether 10:1b:54:xx:xx:d3 brd ff:ff:ff:ff:ff:ff
     inet 192.168.1.100/24 brd 192.168.1.255 scope global endvnic
     valid_lft forever preferred_lft forever
     inet6 fe80::xxxx:xxxx:xxxx:f0d3/64 scope linkifconfig
     valid_lft forever preferred_lft forever
     其中 10:1b:54:xx:xx:d3 为MAC地址
     ```

须知

虚拟网口MAC地址说明

Driver包安装后生成的虚拟网口没有固定的MAC地址，Driver包升级或安装后生成随机数作为MAC地址，虚拟网口MAC地址固定格式：10:1b:54:xx:xx:d3，其中第四和第五位为随机数，10:1b:54前三位MAC地址属于华为公司。

Driver包升级或安装后，虚拟网口MAC地址会变化，需要重新进行MAC地址与IP地址的绑定。

d. 修改“/etc/sysconfig/network-scripts/ifcfg-endvnic”内容（请确保该路径下只有ifcfg-endvnic一个文件，备份文件或重命名文件例如ifcfg-endvnic1，都会影响添加网卡的结果）。

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
#BOOTPROTO=dhcp  #注释掉
DEFROUTE=yes
#IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=endvnic
UUID=6c2db480-1993-40ba-bc21-3541ae728719
DEVICE=endvnic
IPADDR=192.168.1.100   #配置为Host侧虚拟网卡地址
ONBOOT=yes
HWADDR="10:1b:54:xx:xx:d3"   #配置虚拟网卡MAC地址
其中，HWADDR为MAC地址
```

e. 使用**ifup endvnic**命令添加网卡。

f. 执行如下命令，查看网络配置是否成功。

ifconfig

若出现如下回显，表示网络配置成功。否则，请联系华为技术支持处理。

```
endvnic: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.100  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::xxxx:xxxx:xxxx:78d3  prefixlen 64  scopeid 0x20<link>
        ether 10:1b:54:xx:xx:d3  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 42  bytes 2940 (2.9 KB)
        TX errors 8  dropped 8 overruns 0  carrier 0  collisions 0
```

步骤3 登录Device。通过**ssh HwHiAiUser@*xxx*** 命令登录device（xxx为device侧IP地址，如：192.168.1.199）。

说明

- Device重启后，Device侧的配置文件会恢复默认配置（默认账号的超期时间恢复为90天），若用户需要自定义过期时间，可以修改配置文件“/etc/login.defs”中有效期的天数；或者建议用户在自定义的启动脚本中添加对应的chage命令，重新配置过期时间，详情请参见**8.3.3 设置用户有效期**。
- 首次登录Device侧需要修改密码，默认用户名和密码具体请参见**《Atlas硬件产品 用户清单》**。新密码设置规则：密码必须包含字母、数字、特殊字符（特殊字符不能用**@**、**%**、**空格**），长度大于等于8位。

----结束

版权所有 © 华为技术有限公司

9.9.4 修改用户密码

本章节内容仅适用于NPU 1.X.X系列版本。

修改普通操作系统用户的密码

普通操作系统用户（除root）的密码修改方法相同，本节以修改MindStudio安装用户（例如：ascend用户）的密码为例。

步骤1 以普通用户（例如：MindStudio安装用户ascend）登录到操作系统。

步骤2 执行**passwd**命令修改当前用户的密码，例如：修改ascend用户的密码。

```
~$ passwd
```

步骤3 按提示输入新密码后，按“Enter”键。

步骤4 按提示再次输入新密码后，按“Enter”键，完成密码修改。

步骤5 检查新密码是否生效。

1. 退出操作系统。
2. 以新密码重新登录操作系统。

   如果登录成功，表示密码生效。

----结束

修改 root 用户密码

步骤1 以普通用户（例如：MindStudio安装用户ascend）登录到操作系统。

步骤2 执行如下命令切换到root用户。

```
su - root
```

步骤3 执行如下命令修改root用户密码。

```
passwd
```

根据回显信息的提示依次输入普通用户的登录密码、root用户的新密码、root用户的新密码，回车后，密码修改成功。

----结束

9.9.5 通过 PuTTY 登录（网口方式）

操作场景

该任务指导用户使用PuTTY通过局域网远程访问服务器，对服务器实施配置、维护操作。

说明

- 您可以访问chiark网站主页下载PuTTY软件。
- 低版本的PuTTY软件可能导致登录服务器系统失败，建议使用最新版本的PuTTY软件。

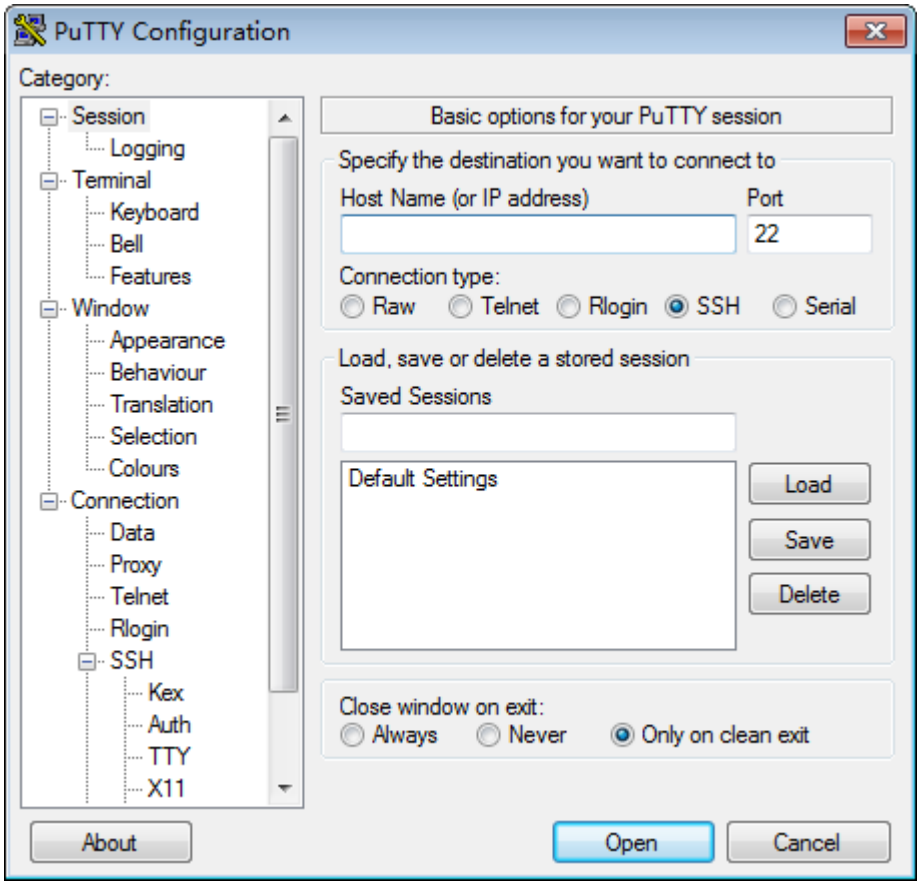
操作步骤

步骤1 设置本地PC的IP地址、子网掩码或者路由，使本地PC能和服务器网络互通。

步骤2 在本地PC上，双击“PuTTY.exe”。

弹出“PuTTY Configuration”窗口。

图 9-4 PuTTY Configuration



步骤3 在左侧导航树中，选择“Session”。

步骤4 设置登录参数。

参数说明如下：

- Host Name (or IP address)：输入要登录服务器的IP地址，如“192.168.2.10”。
- Port：默认设置为“22”。
- Connection type：默认选择“SSH”。
- Close window on exit：默认选择为“Only on clean exit”。

说明

配置“Host Name”后，再配置“Saved Sessions”并单击“Save”保存，则后续使用时直接双击“Saved Sessions”下保存的记录即可登录服务器。

步骤5 单击“Open”。

进入“PuTTY”运行界面，提示“login as:”，等待用户输入用户名。

版权所有 © 华为技术有限公司

说明

- 如果首次登录该目标服务器，则会弹出“PuTTY Security Alert”窗口。单击“是”表示信任此站点，进入“PuTTY”运行界面。
- 如果帐号输入错误，必须重新连接PuTTY。

步骤6 按提示分别输入用户名和密码。

登录完成后，命令提示符左侧显示出当前登录服务器的主机名。

----结束

9.9.6 （可选）源码编译

run安装包二进制交付件的版本与用户环境版本不一致时，用户需要重新进行源码编译。用户环境配置文件：gcc/glibc/kernel/kernel。

源码编译包括如下两种情况。

- driver驱动源码，这部分源码支持自动编译和dkms托管。请参见**9.5.3 源码编译安装**。
- 驱动和用户态程序源码包，这部分源码以tar包形式存放在run安装包中，用户根据自己需要进行解压和编译。请参见**9.9.6.1 源码包编译**。

9.9.6.1 源码包编译

如果host的内核版本与要求的版本差异较大，则采用源码包方式进行编译。源码包编译方式仅适用于NPU 20.0.0以下版本。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

9.9.6.1.1 获取和解压安装包

前提条件

源码编译所有操作都在root用户下执行。

操作步骤

步骤1 获取run包，详细信息请参见**9.3 获取软件包**。

步骤2 安装run安装包到默认路径或指定路径，这里以默认路径“/usr/local/HiAI/”为例。run安装包安装流程请参见**9.5 安装驱动**。

解压后的压缩包的目录结构如**图9-5**所示。

版权所有 © 华为技术有限公司

图 9-5 解压后压缩包的目录结构

```
root@SZX1000481122:/usr/local# tree HiAI/ -L 1
HiAI/
├── driver
├── firmware
├── host_servers_remove.sh
├── host_sys_init.sh
├── host_sys_stop.sh
├── runtime
├── source
├── toolchain
├── uninstall.sh
└── version.info

5 directories, 5 files
root@SZX1000481122:/usr/local# tree HiAI/source/ -L 1
HiAI/source/
├── install.sh
├── readme.txt
└── source.tar.gz

0 directories, 3 files
root@SZX1000481122:/usr/local#
```

说明

- install.sh：源码编译完成后，执行该脚本，把编译的文件拷贝到run安装包安装路径下。
- readme.txt：编译说明文档。
- source.tar.gz：源码包。

----结束

9.9.6.1.2 源码编译

源码编译是可选的，如果run安装包二进制交付件的版本与用户环境不兼容，用户需要重新编译，否则不需要重新编译。用户环境配置文件：gcc/glibc/kernel/kernel。

前提条件

- 源码编译前需要安装如下软件，但不限于如下软件，用户根据编译提示进行安装。
  - Ubuntu操作系统（如下为一条命令，由于PDF文档格式的限制，超过单行的命令，会自动换行，请将命令复制到word或者记事本中，在换行处加入空格，再合成一行复制到服务器中执行）：
    ```
    apt-get install gcc linux-headers-$(uname -r) g++ texinfo flex bison autoconf automake libtool libsysfs-dev
    ```
  - Centos操作系统（如下为一条命令，由于PDF文档格式的限制，超过单行的命令，会自动换行，请将命令复制到word或者记事本中，在换行处加入空格，再合成一行复制到服务器中执行）：
    ```
    yum install gcc linux-headers-$(uname -r) g++ texinfo flex bison autoconf automake libtool libsysfs
    ```
- 权限处理

版权所有 © 华为技术有限公司

run包安装完成后，出于安全考虑，会执行加固操作，如果执行某些操作时，提示没有权限，则参见如下方式解固，操作执行完毕后，再执行加固操作。

请切换到root用户执行如下命令：

解固命令：

```
chattr -i -R run包安装路径
```

加固命令：

```
chattr +i -R run包安装路径
```

例如，run包使用默认安装路径，则解固命令为：**chattr -i -R /usr/local/HiAI/**

操作步骤

源码编译需解压源码包再执行编译，并指定编译工具链和kernel源码路径，具体操作步骤如下：

步骤1 执行**cd /usr/local/HiAI/source**命令，进入源码压缩包路径。

步骤2 执行**tar zxvf source.tar.gz**命令，解压源代码source.tar.gz。

步骤3 执行**cd source**命令，进入源码路径。

步骤4 该操作步骤只对Ubuntu操作系统适用：

配置用户邮箱以及用户名，命令为：

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

步骤5 输入源码编译命令**make ARCH=***?* **KERNEL_DIR=***?* **O=***?*。

参数说明如下：

- ARCH：编译平台，默认为arm64。
- KERNEL_DIR：编译驱动ko指定kernel的根目录，此目录需要有.config文件和include目录。
- O：编译结果输出目录，支持绝对路径与相对路径，默认为当前路径下out文件夹。

举例说明：

- Ubuntu x86操作系统的编译命令（如下为一条命令，由于PDF文档格式的限制，超过单行的命令会自动换行，请将如下命令手动合成一行后再复制到服务器中执行）：
  ```
  make ARCH=x86 HOST_OS=Ubuntu KERNEL_DIR=/usr/src/linux-headers-`uname -r`
  ```
- CentOS x86操作系统的编译命令：
  ```
  make ARCH=x86 HOST_OS=CentOS KERNEL_DIR=/usr/src/kernels/`uname -r`
  ```
- CentOS Arm操作系统的编译命令（如下为一条命令，由于PDF文档格式的限制，超过单行的命令会自动换行，请将如下命令手动合成一行后再复制到服务器中执行）：
  ```
  make ARCH=arm64 HOST_OS=CentOS KERNEL_DIR=/usr/src/kernels/`uname -r`
  CROSS_COMPILE=/usr/bin/
  ```
  其中“/usr/bin/”是arm环境上本地gcc路径。

说明

- CROSS_COMPILE：交叉编译gcc，交叉编译及在Arm环境编译时，需要指定。
- HOST_OS：编译镜像运行的操作系统，目前支持CentOS、Ubuntu（默认）。
- “uname -r”表示使用当前运行的内核版本进行源码编译，在执行编译命令前，可先执行**make clean**清除上次make命令产生的文件，然后再执行编译命令。

步骤6 若出现如图9-6所示界面，则说明编译成功。

图 9-6 编辑结果

```
copy out/host/ide_cmd.sh to out/release_imags/ide_cmd.sh
copy out/host/libc_sec.so to out/release_imags/libc_sec.so
copy out/host/libssl.so.1.1 to out/release_imags/libssl.so.1.1
copy out/host/libcrypto.so.1.1 to out/release_imags/libcrypto.so.1.1
copy out/host/libprotobuf.so.15 to out/release_imags/libprotobuf.so.15
copy out/host/libmmpa.so to out/release_imags/libmmpa.so
copy out/host/libmatrix.so to out/release_imags/libmatrix.so
copy out/host/libmemory.so to out/release_imags/libmemory.so

### build release_modules modules successfully ###

### build successfully ###
root@ubuntu:/usr/local/HiAI/source/source# ls
```

----结束

9.9.6.1.3 源码编译安装

前提条件

完成安装包的解压和编译，详情请参见9.9.6.1.2 源码编译。

操作步骤

步骤1 进入cd /usr/local/HiAI/source目录，执行**install.sh**脚本完成编译文件安装到run安装包目录。

安装命令：**./install.sh** *compile_out_dir*

说明

compile_out_dir为源码编译时的结果输出目录。

- 如果编译时包含O参数，则这里的路径与O参数后的路径一致。
- 如果使用默认路径，则这里的路径为代码路径“source/out/release_imags”。

若出现如下提示信息，则说明安装成功：

```
compiler files install success
```

步骤2 执行**reboot命令**，复位host操作系统。

----结束

9.10 常见问题

版权所有 © 华为技术有限公司

9.10.1 Firmware 固件升级失败

本章节内容仅适用于NPU 1.X.X系列版本。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

问题描述 1

Firmware固件升级时显示：

```
errno 536871427
```

解决方案 1

其他情况一般为用户设置了LD_LIBRARY_PATH环境变量，且该环境变量指向“/root/host”等先前rar包存放lib库的目录。可按照如下操作解决。

步骤1 执行**echo $LD_LIBRARY_PATH**，查看该环境变量。

步骤2 如果设置了LD_LIBRARY_PATH环境变量，执行**export LD_LIBRARY_PATH=**命令进行清除。

步骤3 （可选）进入run安装包升级工具目录“/usr/local/HiAI/firmware/”，执行**ldd upgrade-tool**命令查看工具依赖库链接，此时工具依赖库链接到系统lib库和“/usr/local/HiAI/”目录下的lib库文件。

----结束

问题描述 2

其他情况导致firmware更新失败。

解决方案 2

固件更新依赖host、device上的.ko和.so文件，所以环境异常也会导致固件更新失败。用户可以选择如下方案进行解决。

- 重启host操作系统，复位后重新执行run更新命令。
- 卸载run安装包，复位host操作系统，复位后执行**-- --full**安装命令。该操作仅支持B380及以上版本。

9.10.2 CentOS 虚拟网口 IP 丢失

本章节内容仅适用于NPU 1.X.X系列版本。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

问题描述

CentOS中高级服务NetworkManager和基本服务Network的已知冲突问题，会造成虚拟网口的IP丢失。

解决方案

- 用户不使用高级服务NetworkManager时，直接关闭。

  操作方法：在OS提示符下，执行命令**systemctl disable NetworkManager**

- 用户需要使用高级服务NetworkManager，使用专用命令**nmcli**配置网口，生成对应的ifcfg文件，执行**reboot**命令复位系统后使用**ifconfig**配置网口。

  专用命令**nmcli**命令如下：

  nmcli connection add autoconnect no type ethernet con-name *nic_name* **ifname** *nic_nameip_address*;

  nmcli con up *nic_name*

9.10.3 安装驱动过程中出现“drv_dkms_env_check failed, details in : /var/log/HiAI-install.log”的报错

本章节内容仅适用于NPU 1.X.X系列版本。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

问题描述

安装驱动过程中出现“drv_dkms_env_check failed, details in : /var/log/HiAI-install.log”的报错，请查看日志信息，日志中显示“The program 'dkms' is currently not installed.”，说明为缺少DKMS，请安装DKMS。这里以在Ubuntu系统下为例安装DKMS。

解决方案

服务器联网情况下，通过sudo apt-get install dkms进行安装。

9.10.4 升级系统内核版本时未卸载 run 包导致 run 包不可用

问题描述

如果在安装了run包的环境直接升级内核，再次重启时环境会出现启动异常，如图9-7所示。如果后续在该环境上再安装run包也无法使用。这是由于安装run包时驱动ko加载到内核，而升级内核版本会触发根文件系统打包，会将原来环境中ko打包到根文件系统，导致启动异常。

图 9-7 run 包不可用

```
[    3.690421] i8042: No controller found
[    5.850142] systemd[1]: Failed to start Load Kernel Modules.
[FAILED] Failed to start Load Kernel Modules.
See 'systemctl status systemd-modules-load.service' for details.
[  OK  ] Started Create Static Device Nodes in /dev.
         Starting Apply Kernel Variables...
[  OK  ] Started Apply Kernel Variables.
[  OK  ] Started Setup Virtual Console.
[  OK  ] Started Journal Service.
[  OK  ] Started dracut cmdline hook.
         Starting dracut pre-udev hook...
[  OK  ] Started dracut pre-udev hook.
         Starting udev Kernel Device Manager...
[  OK  ] Started udev Kernel Device Manager.
         Starting udev Coldplug all Devices...
         Mounting Configuration File System...
[  OK  ] Mounted Configuration File System.
[    6.861746] megaraid_sas 0000:18:00.0: IOC INIT command return status SUCCESS for SCSI host 0
[  OK  ] Started udev Coldplug all Devices.
         Starting Show Plymouth Boot Screen...
[  OK  ] Reached target System Initialization.
         Starting dracut initqueue hook...
[    7.059120] megaraid_sas 0000:18:00.0: return from megasas_get_target_prop 7453 return value 2
[    8.777736]  [drv_pcie] [devdrv_get_boot_mode_flag 390] get boot mode from bios err:0x0
[    8.777978]  [drv_pcie] [devdrv_load_file_copy 457] devdrv_device_driver, unable to open file: /home/bios/davinci_cloud.image
 (-5)
[    8.778341]  [drv_pcie] [devdrv_load_file_trans 632] devdrv_device_driver, /home/bios/davinci_cloud.image file copy err
[    8.778909]  [drv_pcie] [devdrv_load_file 1045] devdrv_device_driver, trans file to agent bios failed
[    8.811736]  [drv_pcie] [devdrv_get_boot_mode_flag 390] get boot mode from bios err:0x0
[    8.811976]  [drv_pcie] [devdrv_load_file_copy 457] devdrv_device_driver, unable to open file: /home/bios/davinci_cloud.image
 (-5)
[    8.812341]  [drv_pcie] [devdrv_load_file_trans 632] devdrv_device_driver, /home/bios/davinci_cloud.image file copy err
[    8.812901]  [drv_pcie] [devdrv_load_file 1045] devdrv_device_driver, trans file to agent bios failed
```

解决方案

步骤1 手动卸载ko。

1. 查询当前系统存在ko列表。

   lsmod|grep drv

2. 将查询到的ko一次全部卸载，多个ko名字之间需要加空格。两个ko的示例如下（ko1，ko2请用实际ko名称替换）。

   rmmod *ko1 ko2*

步骤2 打包根文件系统。

dracut --force

步骤3 重启环境。

reboot

----结束

9.10.5 驱动重编译正常但安装上报 modprobe 失败

问题描述

安装驱动过程中，提示“cannot open share object file：No such file or directory”，打开日志文件“/var/log/ascend_seclog/ascend_install.log”报错如下：

```
[ERROR]drv_devdrv_host modprobe failed
```

解决方案

步骤1 执行以下命令，手动加载驱动模块。

modprobe drv_devdrv_host

系统显示示例如下：

版权所有 © 华为技术有限公司

```
modprobe: ERROR: module 'drv_devdrv_host' is unsupported
modprobe: ERROR: Use --allow-unsupported or set allow_unsupported_modules 1 in
modprobe: ERROR: /etc/modprobe.d/10-unsupported-modules.conf
modprobe: ERROR: could not insert 'drv_devdrv_host': Operation not permitted
```

说明

以上回显中的“/etc/modprobe.d/10-unsupported-modules.conf”的为modprobe的限制配置文件路径。**步骤2**命令中的文件路径实际请根据此步骤的回显结果进行替换。

步骤2 执行以下命令，修改modprobe限制配置文件“10-unsupported-modules.conf”。

vi /etc/modprobe.d/10-unsupported-modules.conf

按“i”编辑文件，将文件中“allow_unsupported_modules”的值“0”修改为“1”。

```
#
# Every kernel module has a flag 'supported'. If this flag is not set loading
# this module will taint your kernel. You will not get much help with a kernel
# problem if your kernel is marked as tainted. In this case you firstly have
# to avoid loading of unsupported modules.
#
# Setting allow_unsupported_modules 1 enables loading of unsupported modules
# by modprobe, setting allow_unsupported_modules 0 disables it. This can
# be overriden using the --allow-unsupported-modules commandline switch.
allow_unsupported_modules 0
```

按“Esc”切换到命令模式，输入**:wq!**保存并退出。

步骤3 卸载驱动软件包。请参见**9.7 卸载驱动**。

步骤4 重新安装驱动软件包。请参见**9.5 安装驱动**。

----结束

9.10.6 Centos7.6 系统宿主机目录挂载容器创建“/var/log/npu/slog/slogd”时，提示：Permission denied

本章节内容仅适用于NPU 20.0.X和NPU 20.1.X系列版本。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

问题描述

Centos7.6系统宿主机目录挂载容器，执行**mkdir -p /var/log/npu/slog/slogd**命令创建“/var/log/npu/slog/slogd”目录时，出现如下提示信息：

```
mkdir: cannot create directory '/var/log/npu/slog/slogd': Permission denied
```

解决方案

建议卸载当前系统版本的docker，安装docker 19.03.13版本。参考**12.2 宿主机目录挂载容器**章节重新安装。

版权所有 © 华为技术有限公司

9.10.7 Centos7.6 系统宿主机目录挂载容器拉起 slogd 时，提示：[WARNING] Init config path failed and continue

本章节内容仅适用于NPU 20.0.X和NPU 20.1.X系列版本。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

问题描述

Centos7.6系统宿主机目录挂载容器，执行**su HwHiAiUser --command "/usr/local/Ascend/driver/tools/slogd &"**命令启动slogd进程时，出现如下提示信息：

```
[WARNING] Init config path failed and continue....
```

解决方案

建议卸载当前系统版本的docker，安装docker 19.03.13版本。参考**12.2 宿主机目录挂载容器**章节重新安装。

9.10.8 Ascend_monitor 通讯异常

本章节内容仅适用于NPU 20.0.X和NPU 20.1.X系列版本。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

问题描述

run包安装完成后重启，Host侧日志会有 sendmsg fail:Connection refused相关报错打印,如**图9-8**所示。

图 9-8 发送信息失败：拒绝连接

```
[ERROR] DRV(11116,sklogd):2020-06-18-11:23:45.097.204 [hardware/dev_plat/app_monitor/appmon_lib/utils.c:403][send_msg:403] sendmsg fail:Connection refused.
[ERROR] DRV(11116,sklogd):2020-06-18-11:23:45.097.228 [hardware/dev_plat/app_monitor/appmon_lib/appmon_client.c:186][appmon_client_register:186] send_msg failed ret = 2.
[ERROR] DRV(11115,slogd):2020-06-18-11:23:45.097.295 [hardware/dev_plat/app_monitor/appmon_lib/utils.c:403][send_msg:403] sendmsg fail:Connection refused.
[ERROR] DRV(11115,slogd):2020-06-18-11:23:45.097.337 [hardware/dev_plat/app_monitor/appmon_lib/appmon_client.c:186][appmon_client_register:186] send_msg failed ret = 2.
[ERROR] DRV(11113,log-daemon):2020-06-18-11:23:45.097.426 [hardware/dev_plat/app_monitor/appmon_lib/utils.c:403][send_msg:403] sendmsg fail:Connection refused.
[ERROR] DRV(11113,log-daemon):2020-06-18-11:23:45.097.469 [hardware/dev_plat/app_monitor/appmon_lib/appmon_client.c:186][appmon_client_register:186] send_msg failed ret = 2.
[EVENT] HDC(11117,hdcd):2020-06-18-11:23:45.103.416 [hardware/dev_plat/devhdc/hdc_file/hdcd.c:87][start_file_daemon:87] >>> [87] file daemon 0 start
[EVENT] HDC(11117,hdcd):2020-06-18-11:23:45.103.478 [hardware/dev_plat/devhdc/hdc_file/hdcd.c:87][start_file_daemon:87] >>> [87] file daemon 1 start
```

可能原因

设备重启后，监控进程Ascend_monitor还没完全建立好服务端，被监控进程（如：slogd、sklogd、log-daemon）就与监控进程Ascend_monitor建立连接，注册心跳，可能会出现首次注册心跳失败。

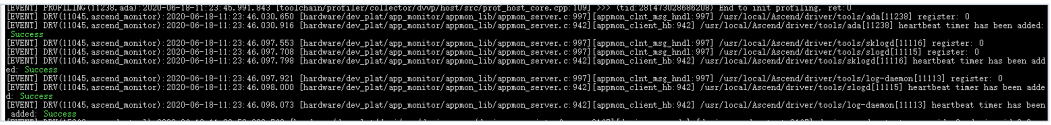
解决方案

针对上述可能原因，可以参考以下方法处理：

步骤1 查看报错信息后续的日志是否出现注册成功的信息，如**图9-9**所示，有则无需处理，对正常功能无影响。

版权所有 © 华为技术有限公司

图 9-9 注册成功



步骤2 如果没有出现注册成功信息，则判断为进程监控通讯异常，需联系华为售后工程师处理。

----结束

9.10.9 安装驱动和固件后重启，“/var/log/npu/slog/host-0”日志中产生“[ERROR] raise script failed, cmd run failed, errno = 0”报错

本章节内容仅适用于NPU 20.0.X和NPU 20.1.X系列版本。

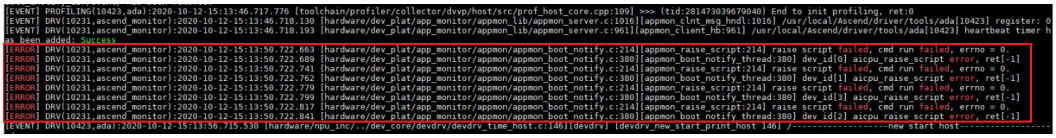
说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

问题描述

使用root用户安装Driver包和Firmware包成功后重启，查看“**/var/log/npu/slog/host-0**”日志会有[ERROR] raise script failed, cmd run failed, errno = 0相关报错打印，如图9-10所示。

图 9-10 日志报错打印



可能原因

由于当前用户权限时间过期，导致驱动日志报错。

解决方案

步骤1 使用root用户，执行以下命令查询用户权限时间。

命令：**chage -l** *用户名*

示例：**chage -l HwHiAiUser**

显示信息如下：

```
bash-4.4$ chage -l HwHiAiUser
Last password change                                    : Apr 15, 2020
Password expires                                        : Jul 14, 2020
Password inactive                                       : Aug 18, 2020
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 90
Number of days of warning before password expires       : 7
bash: EulerOS_history: command not found
```


版权所有 © 华为技术有限公司

步骤2 如果权限时间已经过期，可以执行以下命令通过修改用户密码的方式增加用户权限时间。

命令：**passwd** *用户名*

示例：**passwd HwHiAiUser**

若出现如下信息，表示修改成功。

```
[root@localhost host-0]# passwd HwHiAiUser
Changing password for user HwHiAiUser.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

步骤3 密码修改成功后，再次执行**步骤1**，确认权限时间大于当前时间。如下所示：

```
[root@localhost host-0]# chage -l HwHiAiUser
Last password change                                    : Oct 13, 2020
Password expires                                        : Jan 11, 2021
Password inactive                                       : Feb 15, 2021
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 90
Number of days of warning before password expires       : 7
```

----结束

9.10.10 安装驱动过程中出现“cp: cannot overwrite directory '/usr/local/sbin/npu-smi' with non-directory”的报错

本章节内容仅适用于NPU 20.X.X系列版本。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

问题描述

驱动包安装过程中，出现“cp: cannot overwrite directory '/usr/local/sbin/npu-smi' with non-directory”相关报错打印，如**图9-11**。

图 9-11 安装报错

```
[Driver] [2020-11-17 15:53:28] [INFO]upgradePercentage:10%
cp: cannot overwrite directory '/usr/local/sbin/npu-smi' with non-directory
[Common] [2020-11-17 15:53:29] [ERROR]: driver/tools/npu-smi copy failed!
[Driver] [2020-11-17 15:53:29] [ERROR]Drv_file_copy failed, details in : /var/log/ascend_seclog/ascend_install.log
[Driver] [2020-11-17 15:53:29] [INFO]Install driver failed, please retry after uninstall and reboot!
[Driver] [2020-11-17 15:53:29] [INFO]End time: 2020-11-17 15:53:29
```

可能原因

“/usr/local/sbin”目录下存在“npu-smi”文件夹，驱动安装过程中执行拷贝npu-smi动作失败，导致驱动安装失败。

版权所有 © 华为技术有限公司

解决方案

备份“/usr/local/sbin/npu-smi”文件夹中的内容后，删除“/usr/local/sbin/npu-smi”文件夹，再重新安装驱动。

9.10.11 安装驱动或启动服务过程中出现“please use cmd (lsattr /etc/sudoers) to check permission!”报错

本章节内容仅适用于NPU 20.0.X和20.1.X系列版本。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

问题描述 1

安装或卸载驱动过程中出现“do not have permission, please use cmd (lsattr /etc/sudoers) to check permission!”报错，如图9-12所示：

图 9-12 安装驱动报错

```
[root@localhost zyd]# ./Ascend310-driver-1.73.5.11.50-euleros2.8.aarch64.run --full
Verifying archive integrity...  100%   All good.
Uncompressing ASCEND310 DRIVER RUN PACKAGE  100%
InstallStart: 2021-01-07 13:40:17
InstallLogFile: /var/log/ascend_seclog/ascend_install.log
set username and usergroup, HwHiAiUser:HwHiAiUser
ERR_NO:0x0093;ERR_DES:do not have permission, please use cmd (lsattr /etc/sudoers) to check permission!
InstallEnd: 2021-01-07 13:40:17
```

问题描述 2

启动服务过程中，发现日志文件“/var/log/ascend_seclog/ascend_run_servers.log”存在报错，如图9-13所示：

图 9-13 启动服务报错

```
[root@localhost ascend_seclog]# grep -rn ERROR
ascend_run_servers.log:5:[ERROR]write /etc/sudoers failed, please use cmd (lsattr /etc/sudoers) to check permission!
```

可能原因

“/etc/sudoers”文件不具备写入权限。执行**lsattr /etc/sudoers**命令进行权限确认。

若出现如下回显，表示该文件已被锁定。

```
[root@localhost ~]# lsattr /etc/sudoers
----ia--------e--- /etc/sudoers
[root@localhost ~]#
```

解决方案

执行**chattr -ia /etc/sudoers**命令对“/etc/sudoers”文件进行解锁。

- 对于问题描述1，直接重新安装软件包。若非首次安装场景，软件包安装完成之后需要重启服务器。
- 对于问题描述2，直接重启服务器。

版权所有 © 华为技术有限公司

9.10.12 Ascend 310 芯片无法正常启动，系统日志出现如下打印信息：dev id 0, The address dont comply with the NVME protocol(must be aligned with 4KB).

本章节内容仅适用于NPU 20.2.X及以上系列版本。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

问题描述

Ascend 310芯片无法正常启动时，系统日志有如下打印信息：dev id 0, The address dont comply with the NVME protocol(must be aligned with 4KB).

可能原因

Host内核接口dma_alloc_coherent申请用于Nvme协议的地址不满足4K对齐要求。

根据**内核DMA接口文档**中相关描述，多数平台由于使用一致性内存代价很高，通常最小的分配长度为一个页。

```
Using Consistent DMA mappings
============================

To allocate and map large (PAGE_SIZE or so) consistent DMA regions,
you should do::

    dma_addr_t dma_handle;

    cpu_addr = dma_alloc_coherent(dev, size, &dma_handle, gfp);

where device is a ``struct device *``. This may be called in interrupt
context with the GFP_ATOMIC flag.

Size is the length of the region you want to allocate, in bytes.

This routine will allocate RAM for that region, so it acts similarly to
__get_free_pages() (but takes size instead of a page order).  If your
driver needs regions sized smaller than a page, you may prefer using
the dma_pool interface, described below.

The consistent DMA mapping interfaces, will by default return a DMA address
which is 32-bit addressable.  Even if the device indicates (via the DMA mask)
that it may address the upper 32-bits, consistent allocation will only
return > 32-bit addresses for DMA if the consistent DMA mask has been
explicitly changed via dma_set_coherent_mask().  This is true of the
dma_pool interface as well.

dma_alloc_coherent() returns two values: the virtual address which you
can use to access it from the CPU and dma_handle which you pass to the
card.

The CPU virtual address and the DMA address are both
guaranteed to be aligned to the smallest PAGE_SIZE order which
is greater than or equal to the requested size.  This invariant
exists (for example) to guarantee that if you allocate a chunk
which is smaller than or equal to 64 kilobytes, the extent of the
buffer you receive will not cross a 64K boundary.
```

可能造成不满足4K对齐的原因如下：

版权所有 © 华为技术有限公司

- 用户内核PAGESIZE小于4096字节
- dma_alloc_coherent实现方式与Linux社区不一致

解决方案

步骤1 执行如下命令，查询该环境PAGESIZE。

```
getconf PAGESIZE
4096
```

- 若PAGESIZE小于4096字节，请修改Host内核或修改PAGESIZE。
- 若PAGESIZE大于4096字节，请排查Host内核接口dma_alloc_coherent申请一致性内存方式，确保申请的地址满足4K对齐。

----结束

9.10.13 复位或下电导致 Device 无法正常启动

问题描述 1

在软件包安装或升级过程中，对Device进行了复位或下电操作，出现固件损坏，导致Device无法正常启动。

解决方案 1

1. 对Device连续进行3次复位（非下电，使用**reboot**命令）操作，3次复位之间间隔建议10秒以上。
2. 如Device仍无法启动，则对Device进行下电重启。

问题描述 2

在软件包安装或升级过程中，对Host进行了复位或下电操作，导致升级失败或Device无法正常启动。

解决方案 2

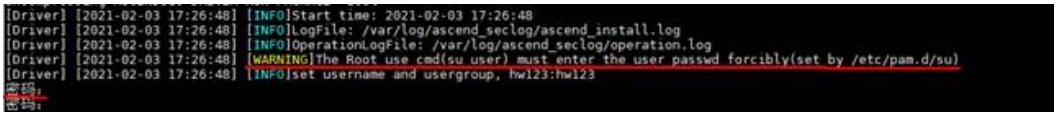
1. 若Device能正常启动，重新升级软件包。
2. 若Device无法正常启动，解决方法请参见**解决方案1**。

9.10.14 root 用户切换到普通用户需要输入密码权限

问题描述

在软件包安装过程中，可能会出现“The Root use cmd(su user) must enter the user passwd forcibly(set by /etc/pam.d/su)”报错。如**图9-14**所示：

图 9-14 报错信息



版权所有 © 华为技术有限公司

可能原因

root用户执行**su**命令进行用户切换时需要输入用户密码。

解决方案

- 直接输入运行用户密码继续完成后续安装。
- 修改系统配置文件“/etc/pam.d/su”，设置root用户执行**su**命令进行用户切换时不需要输入用户密码。操作如下：
  a. 执行如下命令，打开配置文件“/etc/pam.d/su”。
     vi /etc/pam.d/su
  b. 将“auth sufficient pam_rootok.so”前面"**#**"号删除，保存退出。

9.10.15 固件版本不一致导致服务器无法识别 Atlas 300I 推理卡

问题描述

扩容或者备件替换场景下，扩容或替换的Atlas 300I 推理卡内置的固件版本与服务器上已安装的Atlas 300I 推理卡的固件软件版本不一致，导致服务器无法识别扩容或替换的Atlas 300I 推理卡或者Atlas 300I 推理卡在业务运行过程中出现丢卡等问题。

解决方法

升级Atlas 300I 推理卡固件版本以及配套的驱动版本，保证服务器上所有Atlas 300I 推理卡的版本一致；具体升级操作请参见对应版本的《**Atlas 300I 推理卡 NPU驱动和固件升级指南（型号 3000, 3010）**》。

9.10.16 执行 lspci 命令出现“bash: lspci: command not found”报错

问题描述

执行**lspci**命令时，系统提示“bash: lspci: command not found“报错。

可能原因

系统未安装lspci工具。

解决方案

步骤1 执行如下命令查找lspci通过什么安装包提供。

yum whatprovides */lspci

回显如下：

```
pciutils-3.5.1-3.el7.x86_64 : PCI bus related utilities
源    : @anaconda/7.6
匹配来源：
文件名    : /usr/sbin/lspci
```

步骤2 执行如下命令进行安装。

yum install pciutils

步骤3 执行**lspci**命令，若出现如下回显内容，则说明lspci可用。

```
00:00.0 Host bridge: Intel Corporation Sky Lake-E DMI3 Registers (rev 04)
00:04.0 System peripheral: Intel Corporation Sky Lake-E CBDMA Registers (rev 04)
00:04.1 System peripheral: Intel Corporation Sky Lake-E CBDMA Registers (rev 04)
00:04.2 System peripheral: Intel Corporation Sky Lake-E CBDMA Registers (rev 04)
00:04.3 System peripheral: Intel Corporation Sky Lake-E CBDMA Registers (rev 04)
...
```

----结束

9.10.17 查询固件版本号后 Device 侧日志产生“check nve continer fail，ret : 258”报错

本章节内容仅适用于NPU 20.2.X及以上系列版本。

说明

NPU版本号可通过驱动包名称查看，例如驱动包名称为“A300-3000-NPU_Driver-20.0.0-ARM64-Ubuntu18.04.run”，则NPU版本号为20.0.0。

问题描述

安装或升级驱动、固件后，执行*/usr/local/Ascend/***driver/tools/upgrade-tool --device_index -1 --component -1 --version**命令查询固件版本号，命令中的“*/usr/local/Ascend/*”为软件包默认安装路径；使用msnpureport工具收集device日志时发现“messages”目录日志中存在“check nve continer fail，ret : 258”报错。

原因分析

nve固件存放于4个分区，0分区仅用于升级，查询时不属于有效分区，属于功能设计打印信息。

解决方案

属于nve模块设计的合理打印信息，用户无需任何操作。

9.10.18 Centos 8.2 系统最小化安装导致驱动安装失败

问题描述

Centos 8.2+aarch64操作系统安装驱动时出现“DKms install failed, details in : /var/log/ascend_seclog/ascend_install.log”和“Driver_ko_install failed, details in : /var/log/ascend_seclog/ascend_install.log”报错，打开日志文件“/var/log/ascend_seclog/ascend_install.log”查看详情，报错如**图9-15**所示：

图 9-15 编译报错

```
./include/linux/kernel.h:42: note: this is the location of the previous definition
 #define S32_MAX  ((s32)(U32_MAX>>1))

In file included from ./include/uapi/linux/fs.h:13,
                 from ./include/linux/fs.h:44,
                 from ./include/drm-backport/linux/fs.h:14,
                 from /root/selfgz182314442/driver/kernel/devdrv_driver_host/devdrv_devinit.c:12:
./include/linux/limits.h:34: warning: "S64_MAX" redefined
 #define S64_MAX  ((s64)(U64_MAX >> 1))

In file included from ./include/linux/list.h:9,
                 from ./include/linux/rculist.h:10,
                 from ./include/linux/pid.h:5,
                 from ./include/linux/sched.h:14,
                 from ./include/linux/uaccess.h:5,
                 from /root/selfgz182314442/driver/kernel/devdrv_driver_host/devdrv_devinit.c:9:
./include/linux/kernel.h:45: note: this is the location of the previous definition
 #define S64_MAX  ((s64)(U64_MAX>>1))

In file included from ./include/linux/cgroup.h:28,
                 from ./include/linux/memcontrol.h:22,
                 from ./include/linux/swap.h:9,
                 from /root/selfgz182314442/driver/kernel/devdrv_driver_host/devdrv_devinit.c:19:
./include/linux/cgroup-defs.h:541:26: error: 'PATH_MAX' undeclared here (not in a function); did you mean 'PTY_MAX'?
  char release_agent_path[PATH_MAX];
                          ^~~~~~~
                          PTY_MAX
make[3]: *** [scripts/Makefile.build:313: /root/selfgz182314442/driver/kernel/devdrv_driver_host/devdrv_devinit.o] Error 1
make[2]: *** [scripts/Makefile.build:553: /root/selfgz182314442/driver/kernel/devdrv_driver_host] Error 2
make[1]: *** [Makefile:1542: _module_/root/selfgz182314442/driver/kernel] Error 2
make[1]: Leaving directory '/usr/src/kernels/4.18.0-193.el8.aarch64'
make: *** [Makefile:22: all] Error 2
make -C /lib/modules/4.18.0-193.el8.aarch64/build M=/root/selfgz182314442/driver/kernel clean
make[1]: Entering directory '/usr/src/kernels/4.18.0-193.el8.aarch64'
  CLEAN   /root/selfgz182314442/driver/kernel/.tmp_versions
make[1]: Leaving directory '/usr/src/kernels/4.18.0-193.el8.aarch64'
```

可能原因

安装Centos 8.2操作系统时，安装模式选择了“Minimal Install”。

解决方案

重新安装Centos 8.2操作系统，安装模式选择“Server with GUI”，并同时勾选“Development Tools”。

9.11 参考

9.11.1 参数说明

run安装包支持一键式安装，用户根据安装需求选择对应参数完成安装，所有参数都是可选参数。

- 使用NPU 20.X.X及以上系列版本软件包，执行**./****.run* **--[options]**命令，完成软件包的安装，详细参数请参见**表9-6**。

  说明

  如果通过**./****.run* **--help**命令查询出的参数未解释在**表9-6**，则说明该参数预留或适用于其他芯片版本，用户无需关注。

- 使用NPU 1.X.X系列版本软件包，执行**./****.run* **--<commands> --<addition options>**命令，完成对软件包的安装，详细参数请参见**表9-7**。

  说明

  addition options为可选参数，可以和安装、卸载、升级参数组合使用。

版权所有 © 华为技术有限公司

表 9-6 run 安装包支持的参数说明（NPU 20.X.X 及以上系列版本）

参数	说明
info	显示打包信息。
list	run安装包中的文件列表。
check	run安装包完整性校验。
full	完整安装模式：安装所有文件。 **说明** ● 不指定安装目录：**./****.run* **--full** ● 指定安装目录：**./****.run* **--full --install-path=***安装路径*
run	运行态模式：只安装运行场景需安装的文件。 **说明** ● 不指定安装目录：**./****.run* **--run** ● 指定安装目录：**./****.run* **--run --install-path=***安装路径*
docker	容器场景下安装，仅用于driver子包，其他子包默认按full模式。 **说明** ● 不指定安装目录：**./****.run* **--docker** ● 指定安装目录**./****.run* **--docker --install-path=***安装路径*
install-username=<username>	首次安装：可以指定运行用户名，若不指定，则默认是HwHiAiUser。 覆盖安装：沿用上次运行用户名。 **说明** ● 使用NPU 20.0.0版本软件包安装时，不支持指定非HwHiAiUser。 ● Firmware安装不支持指定运行用户名和用户组，共用Driver的运行用户名和用户组。 ● 该参数需要配合“--install-usergroup=<usergroup>”一起使用，并且username需要与创建的用户（**9.4 创建运行用户**）保持一致。
install-usergroup=<usergroup>	首次安装：可以指定运行用户组，若不指定，则默认是HwHiAiUser。 覆盖安装：沿用上次运行用户组。 **说明** ● 使用NPU 20.0.0版本软件包安装时，不支持指定非HwHiAiUser。 ● 该参数需要配合“--install-username=<username>”一起使用，并且usergroup需要与创建的用户组（**9.4 创建运行用户**）保持一致。
install-path=<path>	指定安装目录（该目录仅支持英文字符（a-z、A-Z）、数字（0-9）、.、下划线（_）、中划线（-））， 如不指定： 默认安装路径为：/usr/local/Ascend 若通过该参数指定了安装目录，运行用户需要对指定的安装路径有可读写权限。 **说明** Firmware安装不支持指定路径，共用Driver的安装路径。

版权所有 © 华为技术有限公司

参数	说明
install-for-all	安装或升级时，允许其他用户具有安装群组的权限。 当安装或者升级携带该参数时，软件包中创建的目录及文件，其他用户权限=安装群组权限。 该参数需要与--run、--devel、--full、--upgrade等其中一个参数配合使用，例如：**./*.run** --full --install-for-all **说明** 使用NPU 20.0.0以上、NPU 20.1.X和NPU 21.0.X系列版本软件包安装时支持该参数，但该功能不可用。使用NPU 20.2.X系列版本软件包安装时无此参数。 Firmware不支持该参数。 使用该参数将会存在安全风险：其他所有用户都有权限访问安装目录，请谨慎使用。
reset	恢复初始化配置（该参数只适用于Firmware子包）： ● --full --reset参数通过安装方式恢复初始化配置，需立即重启生效；若安装失败，则恢复初始化配置失败。 ● --upgrade --reset参数通过升级方式恢复初始化配置，需立即重启生效；若升级失败，则恢复初始化配置失败。 恢复初始化配置后，原来用户对device侧的个人配置数据会失效，需要用户重新配置。 **说明** ● --reset参数只支持--full --reset与--upgrade --reset两种模式的使用方法，不支持与其它参数配合使用，也不支持单独使用。 ● 带有--reset参数命令执行完成后，重启生效前，禁止通过ssh、scp等方式访问Device侧的动作，否则会造成恢复初始化配置失败。
devel	开发态模式：包含用户开发时需要用到的头文件。 Firmware子包不支持该参数。
upgrade	升级到当前版本。 **说明** 在安装/升级完成后，需要手动复位OS生效。
uninstall	卸载当前版本。
noexec	不执行安装脚本。配套--extract=*path*使用。格式为：--noexec --extract=*path*
extract=*path*	解压缩安装包中文件到指定目录。
tar arg1 [arg2 …]	对安装包执行tar命令，使用tar后面的参数作为命令的参数。例如执行**--tar xvf**命令，解压run安装包的内容到当前目录。
quiet	静默安装，跳过交互式信息。源码驱动包需要输入指定内核目录安装的情况下，不支持静默安装。 使用该参数进行安装时，若安装目录权限大于755，则会提示安全风险，用户自行选择终止安装或选择其他权限为755的安装目录重新安装。 该参数需要与--run、--devel、--full、--upgrade其中一个参数配合使用，例如**./*.run** --full --quiet

版权所有 © 华为技术有限公司

参数	说明
version	查询版本号。

表 9-7 run 安装包支持的参数说明（NPU 1.X.X 系列版本）

分类	名称	说明
commands	info	显示打包信息。
	list	run安装包中的文件列表。
	check	run安装包完整性校验。
	basic	安装/升级如下模块：nputool，Driver，Runtime。
	classic	安装/升级如下模块：nputool，Driver，Runtime，Toolkit。
	full	安装/升级如下模块：nputool，Firmware，Driver，Runtime，Toolkit，Sample。
	uninstall	卸载当前版本。
	upgrade	升级到当前版本。 **说明** 在安装/升级完成后，需要手动复位OS生效。
	delayedupgrade	升级到当前版本，延时生效。
	viewlog	查看安装日志并退出安装流程。
addition options	path	安装包的路径。 **说明** ● 如果指定run安装包的安装路径，该路径必须是已存在的绝对路径。 ● 如果不指定run安装包的安装路径，则安装到默认路径“/usr/local/HiAI/”下。
	reset	恢复device侧的登录密码。 **说明** ● 此参数和full、upgrade参数组合使用，用于恢复device侧的登录密码。

10 工具

10.1 npu-smi工具

10.2 NPU一键式日志收集工具

10.3 驱动一键式收集日志工具

10.1 npu-smi 工具

10.1.1 安装 npu-smi 工具

run包驱动安装过程中会默认安装npu-smi工具。使用NPU 1.X.X系列版本和NPU 20.0.0版本软件包安装完成后，npu-smi放置在“/usr/local/sbin/”路径下；使用NPU 20.0.0以上、NPU 20.1.X及以上系列版本软件包安装完成后，npu-smi放置在“/usr/local/bin/”和“/usr/local/sbin/”路径下。

使用NPU 1.X.X系列版本软件包安装时，npu-smi工具的rpm包支持单独安装，本章主要介绍rpm包的安装操作。如果npu-smi工具已经安装，则需要选择升级命令进行升级操作。升级操作可参见《**Atlas 300I 推理卡 1.0.0.SPC200及以上 NPU驱动和固件升级指南（型号 3000, 3010）**的“升级npu-smi工具”章节内容。

前提条件

已安装Atlas 300I 推理卡（型号 3000）驱动。具体操作请参见**9.5.2.2 安装驱动（适用于.rpm格式）**。

检查环境

npu-smi工具运行时需要使用“/lib64”目录下的ld-linux-aarch64.so.1文件。如果“/lib64”目录没有该文件，会导致npu-smi工具运行失败。

安装npu-smi工具前，请确保“/lib64”目录下存在ld-linux-aarch64.so.1文件。

若“/lib64”目录下不存在ld-linux-aarch64.so.1文件，请执行如下步骤：

步骤1 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见**9.9.5 通过PuTTY登录（网口方式）**。

步骤2 执行如下命令，切换至root用户。

su - root

步骤3 执行命令**find / -name ld-linux-aarch64.so.1**，查看ld-linux-aarch64.so.1文件所在目录。

- 若不存在该文件，请安装aarch64版本的GNU C库。
- 若存在该文件，执行**步骤4**。

步骤4 执行命令**mkdir -p /lib64**，创建“/lib64”目录。

步骤5 执行命令**ln -s /*xxx*/ld-linux-aarch64.so.1 /lib64**，创建链接，指向ld-linux-aarch64.so.1文件。

说明

xxx表示ld-linux-aarch64.so.1文件所在目录。

----结束

操作步骤

步骤1 参照**9.3 获取软件包**准备npu-smi工具安装包“A800-3000-NPU-EulerOS2.8-ARM-V*x.x.x*.zip”。

步骤2 使用root帐户登录Atlas 300I 推理卡（型号 3000）所在服务器。

步骤3 将“A800-3000-NPU-EulerOS2.8-ARM-V*x.x.x*.zip”上传至Linux系统任意目录下，如/opt下。

步骤4 执行**unzip** *A800-3000-NPU-EulerOS2.8-ARM-Vx.x.x.zip*命令解压安装包，得到rpm安装包“A800-3000-NPU-EulerOS2.8-Dsmitool-ARM-V*x.x.x*.rpm”。

步骤5 执行安装命令完成rpm包安装。

rpm -ivh *A800-3000-NPU-EulerOS2.8-Dsmitool-ARM-Vx.x.x.rpm*

步骤6 执行**npu-smi -v**查询npu-smi工具是否安装成功。

回显如下类似信息，说明安装成功。否则，说明安装失败，请联系华为技术支持处理。

```
npu-smi Version                 : 1.0.2
```

----结束

10.1.2 npu-smi 命令介绍（适用于 5.1.RC 及以上版本）

详细信息请参见**《Atlas 300I 推理卡 npu-smi 命令参考（型号 3000, 3010）》**。

10.1.3 npu-smi 命令介绍（适用于 1.0.11-1.0.15 版本）

10.1.3.1 简介

npu-smi是npu的系统管理工具，可以用于收集设备信息，查看设备健康状态，对设备进行配置以及执行固件升级、清除设备信息等功能。

须知

- 本文“部署场景”中的“Y”表示支持，“N”表示不支持，“NA”表示不涉及。
- 本文中使用实例仅供参考，具体打印信息请以实际环境为准。
- 不支持多线程并发使用npu-smi命令。

10.1.3.2 查询帮助

命令功能

npu-smi -h | --help 用于显示工具的帮助信息。

命令格式

npu-smi -h

参数说明

无

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-1**。

表 10-1 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

显示npu-smi工具的帮助信息。

```
npu-smi -h
 Usage: npu-smi <Command|-h|-v> [Options...]

 npu-smi Command:
      -h          This command's help information
      --help       This command's help information
      -v          Show npu-smi version information
      info         Show hardware details
      set         Modify device configuration properties
      clear        Clear device information
      upgrade       Upgrade MCU firmware
```

输出说明

表 10-2 打印信息说明

字段	说明
-h	命令的帮助信息
--help	命令的帮助信息
-v	npu-smi版本信息
info	显示硬件详细信息
set	修改设备配置属性
clear	清除设备信息
upgrade	升级MCU固件

10.1.3.3 查询 npu-smi 工具版本

命令功能

npu-smi -v用于查询npu-smi工具的版本信息。

命令格式

npu-smi -v

参数说明

无

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-3**。

表 10-3 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询npu-smi的版本信息。

```
npu-smi -v
 npu-smi version: 22.0.0
```

输出说明

表 10-4 打印信息说明

字段	说明
npu-smi version	npu-smi的版本信息

10.1.3.4 信息查询（info）

10.1.3.4.1 查询基本信息

命令功能

npu-smi info 用于查询所有设备的基本信息。

命令格式

npu-smi info

参数说明

无

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-5**。

表 10-5 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询所有设备的基本信息。

```
npu-smi info
+------------------------------------------------------------------------------------------------------+
| npu-smi 21.0.4                    Version: 21.0.4.B030                                               |
+-------------------+-----------------+-------------------------------------------------------------+
| NPU     Name      | Health          | Power(W)          Temp(C)             Hugepages-Usage(page) |
| Chip    Device    | Bus-Id          | AICore(%)         Memory-Usage(MB)                          |
+===================+=================
+==============================================================+
| 6       310       | OK              | 12.8              73            0       / 970              |
| 0       3         | 0000:89:00.0    | 0                 2703 / 8192                              |
+-------------------+-----------------+-------------------------------------------------------------+
| 6       310       | OK              | 12.8              74            0       / 970              |
| 1       4         | 0000:8A:00.0    | 0                 2867 / 8192                              |
+-------------------+-----------------+-------------------------------------------------------------+
| 6       310       | OK              | 12.8              70            0       / 970              |
| 2       5         | 0000:8B:00.0    | 0                 2867 / 8192                              |
+-------------------+-----------------+-------------------------------------------------------------+
| 6       310       | OK              | 12.8              65            0       / 970              |
| 3       6         | 0000:8C:00.0    | 0                 2867 / 8192                              |
+===================+=================
+==============================================================+
```

输出说明

表 10-6 打印信息说明

字段	说明
npu-smi	npu-smi 工具版本
Version	驱动版本
NPU	设备ID
Name	芯片名称
Health	芯片的健康状态 有如下五种状态：OK、Warning、Alarm、Critical或UNKNOWN
Power(W)	芯片功率（单位W）
Temp(C)	芯片温度（单位°C）
Hugepages-Usage(page)	大页占比（单位page），每一个page的大小是2048KB。
Chip	芯片ID
Device	芯片编号
Bus-Id	BUS ID
AICore(%)	AICore占用率
Memory-Usage(MB)	内存占比

字段	说明
注：Hugepages-Usage(page)仅支持NPU 21.0.4及以上系列版本。	

10.1.3.4.2 查询详细信息

命令功能

npu-smi info -t board -i *id* 用于查询设备的详细信息。

命令格式

npu-smi info -t board -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

- 该命令不支持在非root用户下查询Firmware Version和Slot ID。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-7。

表 10-7 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0的具体信息。

```
npu-smi info -t board -i 0
        NPU ID                          : 0
        Product Name                    : IT21DMPB01
        Model                           : NA
        Manufacturer                    : Huawei
        Serial Number                   : 033EFS10J7000021
        Software Version                : 21.0.2.b050
```

```
Firmware Version              : 1.78.23.0.230
Board ID                      : 0xaa
PCB ID                        : A
BOM ID                        : 0
PCIe Bus Info                 : 0000:80:10.0
Slot ID                       : 6
Class ID                      : NA
PCI Vendor ID                 : 0x19E5
PCI Device ID                 : 0xD100
Subsystem Vendor ID           : 0x0200
Subsystem Device ID           : 0x0100
Chip Count                    : 4
Faulty Chip Count             : 0
```

输出说明

表 10-8 打印信息说明

字段	说明
NPU ID	设备ID
Product Name	产品名称
Model	产品型号
Manufacturer	生产厂家
Serial Number	产品序列号
Software Version	驱动版本
Firmware Version	固件版本
Board ID	单板ID
PCB ID	PCB版本编号
BOM ID	BOM版本编号
PCIe Bus Info	PCIe总线信息
Slot ID	槽位信息
Class ID	类型ID
PCI Vendor ID	PCI制造商ID
PCI Device ID	PCI设备ID
Subsystem Vendor ID	子系统制造商ID
Subsystem Device ID	子系统设备ID
Chip Count	芯片个数
Faulty Chip Count	故障芯片个数

10.1.3.4.3 查询指定芯片信息

命令功能

npu-smi info -t board -i *id* **-c** *chip_id* 用于查询设备某一芯片的详细信息。

命令格式

npu-smi info -t board -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

该命令不支持在非root用户下查询NPU的Firmware Version。

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-9**。

表 10-9 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号 3000）	Y	Y	Y	NA

使用实例

查询设备0中编号为0的芯片的详细信息。

```
npu-smi info -t board -i 0 -c 0
        NPU ID                          : 0
        Chip ID                         : 0
        Chip Type                       : Ascend
        Chip Name                       : 310
        Chip Version                    : V100
        Board ID                        : 0xc8
        PCB ID                          : NA
        BOM ID                          : 0
        Die ID                          : V100, 27C56594 02110808 84D1A253 0100A8C0 8D100023
```

```
    Chip Position ID            : 3
    PCIe Bus Info               : 0000:89:00.0
    Firmware Version            : 1.78.23.2.230
```

输出说明

表 10-10 打印信息说明

字段	说明
NPU ID	设备ID
Chip ID	芯片ID
Chip Type	芯片类型
Chip Name	芯片名称
Chip Version	芯片版本
Board ID	*chip_id* 为0/1/2/3时表示NPU类型ID，*chip_id* 为4时表示单板ID
PCB ID	PCB版本编号，取值固定显示为NA
BOM ID	BOM版本编号
Die ID	芯片Die编号
Chip Position ID	芯片位置
PCIe Bus Info	PCIe总线信息
Firmware Version	固件版本

10.1.3.4.4 查询所有芯片监控数据

命令功能

npu-smi info watch 用于查询所有设备上所有芯片的监控数据。

npu-smi info watch -i *id* 用于查询指定设备上所有芯片的监控数据。

命令格式

npu-smi info watch

npu-smi info watch -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-11。

表 10-11 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

- 查询所有设备上所有芯片的监控数据。

```
npu-smi info watch
NpuID(Idx)  ChipId(Idx) Pwr(W)     Temp(C)    AI Core(%)  AI Cpu(%)   Ctrl Cpu(%) Memory(%)
Memory BW(%)
1           0           12.8       47         0           0           0           8           0
1           1           12.8       45         0           0           0           8           0
1           2           12.8       45         0           0           0           8           0
1           3           12.8       44         0           0           0           8           0
2           0           12.8       47         0           0           0           8           0
2           1           12.8       47         0           0           0           9           0
2           2           12.8       46         0           0           0           9           0
2           3           12.8       44         0           0           0           9           0
```

- 查询指定设备上所有芯片的监控数据。

```
npu-smi info watch -i 0
NpuID(Idx)  ChipId(Idx) Pwr(W)     Temp(C)    AI Core(%)  AI Cpu(%)   Ctrl Cpu(%) Memory(%)
Memory BW(%)
0           0           12.8       78         0           0           0           7           0
0           1           12.8       77         0           0           0           7           0
0           2           12.8       76         0           0           0           7           0
0           3           12.8       77         0           0           0           7           0
```

输出说明

表 10-12 打印信息说明

字段	说明
NpuID(Idx)	设备ID
ChipId(Idx)	芯片ID
Pwr(W)	功率（单位W）
Temp(C)	温度（单位°C）

字段	说明
AI Core(%)	AI Core占用率
AI Cpu(%)	AI Cpu占用率
Ctrl Cpu(%)	Ctrl Cpu占用率
Memory(%)	内存占用率
Memory BW(%)	内存带宽占用率

10.1.3.4.5 查询指定芯片监控数据

命令功能

npu-smi info watch -i *id* **-c** *chip_id* **-d** *delay_seconds* **-s** *watch_type* 用于查询指定芯片的监控数据。

命令格式

npu-smi info watch -i *id* **-c** *chip_id* **-d** *delay_seconds* **-s** *watch_type*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。 该命令不支持指定MCU。
delay_seconds	每轮查询延迟时长，单位为秒。取值范围为【1，100】，默认为1秒，
watch_type	监控类型。 ● p代表功率 ● t代表温度 ● a代表AI Core占用率 ● i代表AI Cpu占用率 ● c代表Ctrl Cpu占用率 ● m代表内存占用率 ● b代表内存带宽占用率

使用指南

- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-13。

表 10-13 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0编号为0的芯片的监控数据。

```
npu-smi info watch -i 0 -c 0 -d 5 -s ptaicmb
NpuID(Idx)  ChipId(Idx) Pwr(W)      Temp(C)     AI Core(%)  AI Cpu(%)   Ctrl Cpu(%) Memory(%)
Memory BW(%)
0           0           12.8        70          0           0           0           34          0
0           0           12.8        70          0           0           0           34          0
0           0           12.8        70          0           0           0           34          0
0           0           12.8        70          0           0           0           34          0
```

输出说明

表 10-14 打印信息说明

字段	说明
NpuID(Idx)	设备ID
ChipId(Idx)	芯片ID
Pwr(W)	功率（单位W）
Temp(C)	温度（单位°C）
AI Core(%)	AI Core占用率
AI Cpu(%)	AI Cpu占用率
Ctrl Cpu(%)	Ctrl Cpu占用率
Memory(%)	内存占用率
Memory BW(%)	内存带宽占用率

10.1.3.4.6 查询所有 NPU 设备

命令功能

npu-smi info -l 用于查询所有NPU设备。

命令格式

npu-smi info -l

参数说明

无

使用指南

docker容器中不支持查询product Name，Serial Number域。

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-15**。

表 10-15 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询所有NPU设备的相关信息。

```
npu-smi info -l
      Card Count                    : 1

      NPU ID                        : 6
      Product Name                  : IT21DMPB01
      Serial Number                 : 033EFS10J7000021
      Chip Count                    : 4
```

输出说明

表 10-16 打印信息说明

字段	说明
Card Count	NPU卡个数

字段	说明
NPU ID	设备ID
Product Name	产品名称
Serial Number	产品序列号
Chip Count	芯片个数

10.1.3.4.7 查询所有芯片映射关系信息

命令功能

npu-smi info -m 用于查询所有芯片的映射关系信息。

命令格式

npu-smi info -m

参数说明

无

使用指南

- docker容器中不显示MCU信息。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-17。

表 10-17 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询目前所有芯片的映射关系信息。

```
npu-smi info -m
```

```
NPU ID                  Chip ID                 Chip Logic ID           Chip Name
6                       0                       0                       MiniD
6                       1                       1                       MiniD
6                       2                       2                       MiniD
6                       3                       3                       MiniD
6                       4                       -                       Mcu
```

输出说明

表 10-18 打印信息说明

字段	说明
NPU ID	设备ID
Chip ID	芯片ID
Chip Logic ID	芯片逻辑ID
Chip Name	芯片名称

10.1.3.4.8 查询所有芯片常用信息

命令功能

npu-smi info -t common -i *id* 用于查询所有芯片常用信息。

命令格式

npu-smi info -t common -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-19**。

表 10-19 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户

Atlas 300I 推理卡（型号：3000）	Y	Y	NPU 21.0.2.spc008 之前版本：N NPU 21.0.2.spc008 及以上版本：Y	NA

使用实例

查询设备0中所有芯片的常用信息。

```
npu-smi info -t common -i 0
        NPU ID                         : 0
        Chip Count                     : 4

        Chip ID                        : 0
        Memory Usage Rate(%)           : 33
        Aicore Usage Rate(%)           : 0
        Temperature(C)                 : 68
        Rated Power Dissipation(W)     : 12.8

        Chip ID                        : 1
        Memory Usage Rate(%)           : 35
        Aicore Usage Rate(%)           : 0
        Temperature(C)                 : 69
        Rated Power Dissipation(W)     : 12.8

        Chip ID                        : 2
        Memory Usage Rate(%)           : 35
        Aicore Usage Rate(%)           : 0
        Temperature(C)                 : 65
        Rated Power Dissipation(W)     : 12.8

        Chip ID                        : 3
        Memory Usage Rate(%)           : 35
        Aicore Usage Rate(%)           : 0
        Temperature(C)                 : 60
        Rated Power Dissipation(W)     : 12.8

        Chip Name                      : MCU
        Temperature(C)                 : 56
        Power Dissipation(W)           : 18.0
```

输出说明

表 10-20 打印信息说明

字段	说明
NPU ID	设备ID
Chip Count	芯片个数
Chip ID	芯片ID
Memory Usage Rate(%)	内存占用率

字段	说明
Aicore Usage Rate(%)	Aicore占用率
Temperature(C)	温度（单位°C）
Chip Name	芯片名称
Power Dissipation(W)	功率（单位W）
Rated Power Dissipation(W)	额定功率（单位W）

10.1.3.4.9 查询所有芯片闪存信息

命令功能

npu-smi info -t flash -i *id* 用于查询所有芯片的闪存信息。

命令格式

npu-smi info -t flash -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-21。

表 10-21 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号 3000）	Y	Y	Y	NA

使用实例

查询设备0中所有芯片的闪存信息。

```
npu-smi info -t flash -i 0
        NPU ID                          : 0
        Chip Count                      : 4

        Chip ID                         : 0
        Flash Count                     : 1

        Flash ID                        : 624427202
        Manufacturer ID                 : 0xC2
        Capacity(MB)                    : 16

        Chip ID                         : 1
        Flash Count                     : 1

        Flash ID                        : 624427202
        Manufacturer ID                 : 0xC2
        Capacity(MB)                    : 16

        Chip ID                         : 2
        Flash Count                     : 1

        Flash ID                        : 624427202
        Manufacturer ID                 : 0xC2
        Capacity(MB)                    : 16

        Chip ID                         : 3
        Flash Count                     : 1

        Flash ID                        : 624427202
        Manufacturer ID                 : 0xC2
        Capacity(MB)                    : 16
```

输出说明

表 10-22 打印信息说明

字段	说明
NPU ID	设备ID
Chip Count	芯片个数
Chip ID	芯片ID
Flash Count	Flash个数
Flash ID	Flash ID
Manufacturer ID	制造商ID
Capacity(MB)	容量（单位MB）

10.1.3.4.10 查询指定芯片闪存信息

命令功能

npu-smi info -t flash -i *id* **-c** *chip_id* 用于查询某个芯片的闪存信息。

命令格式

npu-smi info -t flash -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。 该命令不支持指定MCU。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-23**。

表 10-23 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0中编号为0的芯片的闪存信息。

```
npu-smi info -t flash -i 0 -c 0
    Flash Count                   : 1

    Flash ID                      : 624427202
    Manufacturer ID               : 0xC2
    Capacity(MB)                  : 16
```

输出说明

表 10-24 打印信息说明

字段	说明
Flash Count	Flash个数

字段	说明
Flash ID	Flash ID
Manufacturer ID	制造商ID
Capacity(MB)	容量（单位MB）

10.1.3.4.11 查询所有芯片内存信息

命令功能

npu-smi info -t memory -i *id* 用于查询所有芯片的内存信息。

命令格式

npu-smi info -t memory -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-25**。

表 10-25 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0的所有芯片的内存信息。

```
npu-smi info -t memory -i 0
        NPU ID                    : 0
        Chip Count                : 4
```

```
    Chip ID                    : 0
    Capacity(MB)               : 8192
    Clock Speed(MHz)           : 1600

    Chip ID                    : 1
    Capacity(MB)               : 8192
    Clock Speed(MHz)           : 1600

    Chip ID                    : 2
    Capacity(MB)               : 8192
    Clock Speed(MHz)           : 1600

    Chip ID                    : 3
    Capacity(MB)               : 8192
    Clock Speed(MHz)           : 1600
```

输出说明

表 10-26 打印信息说明

字段	说明
NPU ID	设备ID
Chip Count	芯片个数
Chip ID	芯片ID
Capacity(MB)	容量（单位MB）
Clock Speed(MHz)	时钟频率（单位MHZ）

10.1.3.4.12 查询指定芯片内存信息

命令功能

npu-smi info -t memory -i *id* **-c** *chip_id* 用于查询某个芯片的内存信息。

命令格式

npu-smi info -t memory -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。 该命令不支持指定MCU。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-27。

表 10-27 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0中编号为0的芯片的内存信息。

```
npu-smi info -t memory -i 0 -c 0
        Capacity(MB)                : 8192
        Clock Speed(MHz)            : 1600
```

输出说明

表 10-28 打印信息说明

字段	说明
Capacity(MB)	容量（单位MB）
Clock Speed(MHz)	时钟频率（单位MHZ）

10.1.3.4.13 查询所有芯片统计信息

命令功能

npu-smi info -t usages -i *id* 用于查询所有芯片的统计信息。

命令格式

npu-smi info -t usages -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-29**。

表 10-29 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0所有芯片的统计信息。

```
npu-smi info -t usages -i 0
        NPU ID                         : 0
        Chip Count                     : 4

        Chip ID                        : 0
        Memory Capacity(MB)            : 8192
        Memory Usage Rate(%)           : 38
        Hugepages Total(page)          : 970
        Hugepages Usage Rate(%)        : 0
        Aicore Usage Rate(%)           : 50
        Aicpu Usage Rate(%)            : 3
        Ctrlcpu Usage Rate(%)          : 0
        Memory Bandwidth Usage Rate(%) : 13

        Chip ID                        : 1
        Memory Capacity(MB)            : 8192
        Memory Usage Rate(%)           : 40
        Hugepages Total(page)          : 970
        Hugepages Usage Rate(%)        : 0
        Aicore Usage Rate(%)           : 0
        Aicpu Usage Rate(%)            : 1
        Ctrlcpu Usage Rate(%)          : 0
        Memory Bandwidth Usage Rate(%) : 4

        Chip ID                        : 2
        Memory Capacity(MB)            : 8192
        Memory Usage Rate(%)           : 39
        Hugepages Total(page)          : 970
        Hugepages Usage Rate(%)        : 0
        Aicore Usage Rate(%)           : 0
```

```
    Aicpu Usage Rate(%)            : 1
    Ctrlcpu Usage Rate(%)          : 0
    Memory Bandwidth Usage Rate(%) : 3

    Chip ID                        : 3
    Memory Capacity(MB)            : 8192
    Memory Usage Rate(%)           : 39
    Hugepages Total(page)          : 970
    Hugepages Usage Rate(%)        : 0
    Aicore Usage Rate(%)           : 55
    Aicpu Usage Rate(%)            : 0
    Ctrlcpu Usage Rate(%)          : 0
    Memory Bandwidth Usage Rate(%) : 13
```

输出说明

表 10-30 打印信息说明

字段	说明
NPU ID	设备ID
Chip Count	芯片个数
Chip ID	芯片ID
Memory Capacity(MB)	内存容量（单位MB）
Memory Usage Rate(%)	内存占用率
Hugepages Total(page)	大页容量（单位page），每一个page的大小是2048KB。
Hugepages Usage Rate(%)	大页占用率
Aicore Usage Rate(%)	Aicore占用率
Aicpu Usage Rate(%)	Aicpu占用率
Ctrlcpu Usage Rate(%)	Ctrlcpu占用率
Memory Bandwidth Usage Rate(%)	内存带宽占用率
注：Hugepages Total(page)和Hugepages Usage Rate(%)仅支持NPU 21.0.4及以上系列版本。	

10.1.3.4.14 查询指定芯片统计信息

命令功能

npu-smi info -t usages -i *id* **-c** *chip_id* 用于查询某个芯片的统计信息。

命令格式

npu-smi info -t usages -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。 该命令不支持指定MCU。

使用指南

- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-31**。

表 10-31 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0中编号为0的芯片的统计信息。

```
npu-smi info -t usages -i 0 -c 0
        Memory Capacity(MB)            : 8192
        Memory Usage Rate(%)           : 30
        Hugepages Total(page)          : 970
        Hugepages Usage Rate(%)        : 0
        Aicore Usage Rate(%)           : 5
        Aicpu Usage Rate(%)            : 1
        Ctrlcpu Usage Rate(%)          : 0
        Memory Bandwidth Usage Rate(%) : 12
```

输出说明

表 10-32 打印信息说明

字段	说明
Memory Capacity(MB)	内存容量（单位MB）
Memory Usage Rate(%)	内存占用率

字段	说明
Hugepages Total(page)	大页容量（单位page），每一个page的大小是2048KB。
Hugepages Usage Rate(%)	大页占用率
Aicore Usage Rate(%)	Aicore占用率
Aicpu Usage Rate(%)	Aicpu占用率
Ctrlcpu Usage Rate(%)	Ctrlcpu占用率
Memory Bandwidth Usage Rate(%)	内存带宽占用率
注：Hugepages Total(page)和Hugepages Usage Rate(%)仅支持NPU 21.0.4及以上系列版本。	

10.1.3.4.15 查询所有芯片温度

命令功能

npu-smi info -t temp -i *id* 用于查询所有芯片的温度。

命令格式

npu-smi info -t temp -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

docker容器中不显示MCU信息。

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-33**。

表 10-33 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户

Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0的所有芯片的温度。

```
npu-smi info -t temp -i 0
        NPU ID                         : 0
        Chip Count                     : 4

        Chip ID                        : 0
        Temperature (C)                : 67

        Chip ID                        : 1
        Temperature (C)                : 67

        Chip ID                        : 2
        Temperature (C)                : 64

        Chip ID                        : 3
        Temperature (C)                : 58

        Temperature (C)                : 55
        Chip Name                      : MCU
```

输出说明

表 10-34 打印信息说明

字段	说明
NPU ID	设备ID
Chip Count	芯片个数
Chip ID	芯片ID
Temperature (C)	温度（单位°C）
Chip Name	芯片名称

10.1.3.4.16 查询指定芯片温度

命令功能

npu-smi info -t temp -i *id* **-c** *chip_id* 用于查询某个芯片的温度。

命令格式

npu-smi info -t temp -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-35**。

表 10-35 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0中编号为0的芯片的温度。

```
npu-smi info -t temp -i 0 -c 0
        Temperature (C)                 : 68
```

输出说明

表 10-36 打印信息说明

字段	说明
Temperature (C)	温度（单位°C）

10.1.3.4.17 查询所有芯片功率

命令功能

npu-smi info -t power -i *id* 用于查询所有芯片的功率。

命令格式

npu-smi info -t power -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

docker容器中不显示MCU信息。

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-37**。

表 10-37 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0的所有芯片的功率。

```
npu-smi info -t power -i 0
        NPU ID                          : 0
        Chip Count                      : 4

        Chip ID                         : 0
        Rated Power Dissipation(W)      : 12.8

        Chip ID                         : 1
        Rated Power Dissipation(W)      : 12.8

        Chip ID                         : 2
        Rated Power Dissipation(W)      : 12.8

        Chip ID                         : 3
        Rated Power Dissipation(W)      : 12.8

        Power Dissipation(W)            : 17.0
        Chip Name                       : MCU
```

输出说明

表 10-38 打印信息说明

字段	说明
NPU ID	设备ID
Chip Count	芯片个数
Chip ID	芯片ID
Rated Power Dissipation(W)	额定功率（单位W）
Power Dissipation(W)	实时功率（单位W）
Chip Name	芯片名称

10.1.3.4.18 查询指定芯片功率

命令功能

npu-smi info -t power -i *id* **-c** *chip_id* 用于查询某个芯片的功率。

命令格式

npu-smi info -t power -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-39**。

表 10-39 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户

Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0中编号为0的芯片功率。

```
npu-smi info -t power -i 0 -c 0
      Rated Power Dissipation(W)     : 12.8
```

输出说明

表 10-40 打印信息说明

字段	说明
Rated Power Dissipation(W)	额定功率（单位W）

10.1.3.4.19 查询所有芯片电压

命令功能

npu-smi info -t volt -i *id* 用于查询所有芯片的电压。

命令格式

npu-smi info -t volt -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

docker容器中不显示MCU信息。

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-41**。

表 10-41 部署场景

产品形态	Linux物理机	Linux容器	Windows操作系统

	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0的所有芯片电压信息。

```
npu-smi info -t volt -i 0
        NPU ID                         : 0
        Chip Count                     : 4

        Chip ID                        : 0
        Voltage(V)                     : 0.80

        Chip ID                        : 1
        Voltage(V)                     : 0.80

        Chip ID                        : 2
        Voltage(V)                     : 0.80

        Chip ID                        : 3
        Voltage(V)                     : 0.80

        Voltage(V)                     : NA
        Chip Name                      : MCU
```

输出说明

表 10-42 打印信息说明

字段	说明
NPU ID	设备ID
Chip Count	芯片个数
Chip ID	芯片ID
Voltage(V)	电压值，单位为：伏特(V)
Chip Name	芯片名称

10.1.3.4.20 查询指定芯片电压

命令功能

npu-smi info -t volt -i *id* **-c** *chip_id* 用于查询某个芯片的电压。

命令格式

npu-smi info -t volt -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info --l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-43**。

表 10-43 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0中编号为0的芯片的电压。

```
npu-smi info -t volt -i 0 -c 0
      Voltage(V)                    : 0.80
```

输出说明

表 10-44 打印信息说明

字段	说明
Voltage(V)	电压值，单位：伏特(V)

10.1.3.4.21 查询所有芯片健康状态

命令功能

npu-smi info -t health -i *id* 用于查询所有芯片的健康状态。

命令格式

npu-smi info -t health -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

docker容器中不显示MCU状态。

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-45**。

表 10-45 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0所有芯片的健康状态。

```
npu-smi info -t health -i 0
      NPU ID                    : 0
      Chip Count                : 4

      Chip ID                   : 0
      Health                    : OK

      Chip ID                   : 1
      Health                    : OK

      Chip ID                   : 2
      Health                    : OK

      Chip ID                   : 3
```

```
    Health                  : OK

    Health                  : OK
    Chip Name                 : MCU
```

输出说明

表 10-46 打印信息说明

字段	说明
NPU ID	设备ID
Chip Count	芯片个数
Chip ID	芯片ID
Health	芯片的健康状态 有如下五种状态：OK、Warning、Alarm、Critical或UNKNOWN
Chip Name	芯片名称

10.1.3.4.22 查询指定芯片健康状态

命令功能

npu-smi info -t health -i *id* **-c** *chip_id* 用于查询某个芯片的健康状态。

命令格式

npu-smi info -t health -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-47**。

表 10-47 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0中编号为0的芯片的健康状态。

```
npu-smi info -t health -i 0 -c 0
        Health Status                   : OK
        Error Code                      : NA
        Error Information               : NA
```

输出说明

表 10-48 打印信息说明

字段	说明
Health Status	芯片的健康状态 有如下五种状态：OK、Warning、Alarm、Critical或UNKNOWN
Error Code	错误码
Error Information	错误信息

10.1.3.4.23 查询 MCU 监控信息

命令功能

npu-smi info -t mcu-monitor -i *id* **-c** *chip_id* 用于查询MCU监控信息。

命令格式

npu-smi info -t mcu-monitor -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-49。

表 10-49 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号 3000）	N	N	N	NA

使用实例

查询MCU监控信息。

```
npu-smi info -t mcu-monitor -i 0 -c 1
This device does not support querying mcu-monitor.
```

输出说明

无

10.1.3.4.24 查询所有芯片的 ssh 使能状态

命令功能

npu-smi info -t ssh-enable -i *id* 用于查询所有芯片的ssh使能状态。

命令格式

npu-smi info -t ssh-enable -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-50。

表 10-50 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0所有芯片的ssh使能状态。

```
npu-smi info -t ssh-enable -i 0
        NPU ID                          : 0
        Chip Count                      : 4

        Chip ID                         : 0
        SSH-Enable Status               : False

        Chip ID                         : 1
        SSH-Enable Status               : True

        Chip ID                         : 2
        SSH-Enable Status               : True

        Chip ID                         : 3
        SSH-Enable Status               : True
```

输出说明

表 10-51 打印信息说明

字段	说明
NPU ID	设备ID
Chip Count	芯片个数

字段	说明
Chip ID	芯片ID
SSH-Enable Status	芯片的ssh使能状态 有如下两种状态：False，True

10.1.3.4.25 查询指定芯片的 ssh 使能状态

命令功能

npu-smi info -t ssh-enable -i *id* **-c** *chip_id* 用于查询某个芯片的ssh使能状态。

命令格式

npu-smi info -t ssh-enable -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。 该命令不支持指定MCU。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-52**。

表 10-52 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0中编号为0的芯片的ssh使能状态。

```
npu-smi info -t ssh-enable -i 0 -c 0
  SSH-Enable Status              : False
```

输出说明

表 10-53 打印信息说明

字段	说明
SSH-Enable Status	芯片的ssh使能状态 有如下两种状态：False，True

10.1.3.4.26 查询所有芯片 ECC 错误计数

命令功能

npu-smi info -t ecc -i *id* 用于查询所有芯片的ECC错误计数。

命令格式

npu-smi info -t ecc -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-54**。

表 10-54 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0所有芯片的ECC错误统计。

```
npu-smi info -t ecc -i 0
        NPU ID                          : 0
        Chip Count                      : 4

        Chip ID                         : 0
        Single-bit Error Count                : 0
        Double-bit Error Count                : 0

        Chip ID                         : 1
        Single-bit Error Count                : 0
        Double-bit Error Count                : 0

        Chip ID                         : 2
        Single-bit Error Count                : 0
        Double-bit Error Count                : 0

        Chip ID                         : 3
        Single-bit Error Count                : 0
        Double-bit Error Count                : 0
```

输出说明

表 10-55 打印信息说明

字段	说明
NPU ID	设备ID
Chip Count	芯片个数
Chip ID	芯片ID
Single-bit Error Count	单比特错误数量
Double-bit Error Count	多比特错误数量

10.1.3.4.27 查询指定芯片 ECC 错误计数

命令功能

npu-smi info -t ecc -i *id* **-c** *chip_id* 用于查询某个芯片的ECC错误计数。

命令格式

npu-smi info -t ecc -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

类型	描述
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。 该命令不支持指定MCU。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-56**。

表 10-56 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0中编号为0的芯片的ECC错误计数。

```
npu-smi info -t ecc -i 0 -c 0
        Single-bit Error Count                : 0
        Double-bit Error Count                : 0
```

输出说明

表 10-57 打印信息说明

字段	说明
Single-bit Error Count	单比特错误数量
Double-bit Error Count	多比特错误数量

10.1.3.4.28 查询 ECC 使能状态

命令功能

npu-smi info -t ecc-enable -i *id* 用于查询NPU设备的ECC使能状态。

命令格式

npu-smi info -t ecc-enable -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-58**。

表 10-58 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0的ECC使能状态。

```
npu-smi info -t ecc-enable -i 0
        ECC-Enable Status              : True
```

输出说明

表 10-59 打印信息说明

字段	说明
ECC-Enable Status	DDR的ECC使能状态 有如下两种状态：False、True

10.1.3.4.29 查询 p2p 使能状态

命令功能

npu-smi info -t p2p-enable -i *id* 用于查询NPU设备的p2p使能状态。

命令格式

npu-smi info -t p2p-enable -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-60。

表 10-60 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0的p2p使能状态。

```
npu-smi info -t p2p-enable -i 0
        P2P-Enable Status              : False
```

输出说明

表 10-61 打印信息说明

字段	说明
P2P-Enable Status	p2p使能状态 有如下两种状态：False、True

10.1.3.4.30 查询指定芯片 boot 区

命令功能

npu-smi info -t boot-area -i *id* **-c** *chip_id* 用于查询某个芯片boot区。

命令格式

npu-smi info -t boot-area -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-62**。

表 10-62 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

查询设备0中编号为0的boot区。

```
npu-smi info -t boot-area -i 0 -c 0
This device does not support querying boot-area.
```

输出说明

无

10.1.3.4.31 查询指定芯片传感器信息

命令功能

npu-smi info -t sensors -i *id* **-c** *chip_id* 用于查询指定芯片传感器信息。

命令格式

npu-smi info -t sensors -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-63**。

表 10-63 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

查询设备0编号为0的芯片传感器信息。

```
npu-smi info -t sensors -i 0 -c 0
This device does not support querying sensors.
```

输出说明

无

10.1.3.4.32 查询指定芯片系统时间

命令功能

npu-smi info -t sys-time -i *id* **-c** *chip_id* 用于查询某个芯片的系统时间。

命令格式

npu-smi info -t sys-time -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-64**。

表 10-64 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

查询设备0编号为0的芯片系统时间。

```
npu-smi info -t sys-time -i 0 -c 0
This device does not support querying sys-time.
```

输出说明

无

10.1.3.4.33 查询 NPU 与 MCU 之间的 IIC 通道是否正常

命令功能

npu-smi info -t i2c_check -i *id* 用于查询NPU与MCU之间的IIC通道是否正常。

命令格式

npu-smi info -t i2c_check -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-65**。

表 10-65 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	N	N	NA

使用实例

查询设备0与MCU之间的IIC通道状态。

```
npu-smi info -t i2c_check -i 0
      Checking... Please do not interrupt!
      MiniD 1 to MCU iic channel status: OK
      MiniD 3 to MCU iic channel status: OK
      MCU to miniD 0 iic channel status: OK
      MCU to miniD 1 iic channel status: OK
      MCU to miniD 2 iic channel status: OK
      MCU to miniD 3 iic channel status: OK
```

输出说明

表 10-66 打印信息说明

字段	说明
miniD 1 to MCU iic channel status	NPU设备中MiniD 1与MCU之间的I2C通道状态 有如下三种状态：OK，Fault，Unknown
miniD 3 to MCU iic channel status	NPU设备中MiniD 3与MCU之间的I2C通道状态 有如下三种状态：OK，Fault，Unknown
MCU to miniD 0 iic channel status	NPU设备中MCU与MiniD 0之间的I2C通道状态 有如下三种状态：OK，Fault，Unknown
MCU to miniD 1 iic channel status	NPU设备中MCU与MiniD 1之间的I2C通道状态 有如下三种状态：OK，Fault，Unknown
MCU to miniD 2 iic channel status	NPU设备中MCU与MiniD 2之间的I2C通道状态 有如下三种状态：OK，Fault，Unknown
MCU to miniD 3 iic channel status	NPU设备中MCU与MiniD 3之间的I2C通道状态 有如下三种状态：OK，Fault，Unknown

10.1.3.4.34 查询产品类型

命令功能

npu-smi info -t product -i *id* 用于查询产品类型。

命令格式

npu-smi info -t product -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-67。

表 10-67 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0的产品类型。具体产品型号以实际查询的产品为准。

```
npu-smi info -t product -i 0
        Product Type                    : Atlas 300I Model 3000
```

输出说明

表 10-68 打印信息说明

字段	说明
Product Type	产品类型

10.1.3.4.35 查询指定芯片的产品类型

命令功能

npu-smi info -t product -i *id* **-c** *chip_id* 用于查询指定芯片的产品类型。

命令格式

npu-smi info -t product -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

类型	描述
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。 该命令不支持指定MCU。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-69**。

表 10-69 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0中编号为0的芯片的产品类型。具体产品型号以实际查询的产品为准。

```
npu-smi info -t product -i 0 -c 0
        Product Type                    : Atlas 300I Model 3000
```

输出说明

表 10-70 打印信息说明

字段	说明
Product Type	产品类型

10.1.3.4.36 查询指定芯片的 Matrix 进程监控数据

命令功能

npu-smi info proc -i *id* **-c** *chip_id* **-d** delay_seconds 用于查询指定芯片的Matrix进程监控数据。

命令格式

npu-smi info proc -i *id* **-c** *chip_id* **-d** delay_seconds

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。
delay_seconds	每轮查询延迟时长，单位为秒。默认为1秒，取值范围：1~100。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-71**。

表 10-71 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

查询设备0，编号为0的芯片每隔2s内的Matrix进程监控数据。

```
npu-smi info proc -i 0 -c 0 -d 2
This device does not support querying proc info.
```

输出说明

无

10.1.3.4.37 查询用户证书

命令功能

npu-smi info -t license -i *id* 用于查询用户的证书。

命令格式

npu-smi info -t license -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-72。

表 10-72 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	N	NA

使用实例

获取设备0的证书。

```
npu-smi info -t license -i 0
      License              : testjn
```

输出说明

表 10-73 打印信息说明

字段	说明
License	用户证书

10.1.3.4.38 查询用户自定义信息

命令功能

npu-smi info -t customized-info -i *id* 用于查询用户自定义信息。

命令格式

npu-smi info -t customized-info -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-74**。

表 10-74 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

获取设备0的用户自定义信息。

```
npu-smi info -t customized-info -i 0
This device does not support querying customized-info.
```

输出说明

无

10.1.3.4.39 查询 AI 芯片的 IP 信息

命令功能

npu-smi info -t ip -i *id* **-c** *chip_id* 用于查询AI芯片的IP信息。

命令格式

npu-smi info -t ip -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。 该命令不支持指定MCU。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-75**。

表 10-75 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0中编号为0的AI芯片的IP信息。

```
npu-smi info -t ip -i 0 -c 0
        ip                          : 192.168.5.199
        netmask                     : 255.255.255.0
```

输出说明

表 10-76 打印信息说明

字段	说明
ip	ip地址
netmask	子网掩码

10.1.3.4.40 查询所有芯片的 PCIE 误码计数

命令功能

npu-smi info -t err-count -i *id* 用于查询所有芯片PCIE误码计数。

npu-smi info -t pcie-err -i *id* 用于查询所有芯片PCIE误码计数。

命令格式

npu-smi info -t err-count -i *id*

npu-smi info -t pcie-err -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

npu-smi info -t err-count -i *id* 支持的产品形态、应用场景、用户权限以及操作系统请参见表10-77。

表 10-77 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

npu-smi info -t pcie-err -i *id* 支持的产品形态、应用场景、用户权限以及操作系统请参见表10-78。

表 10-78 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户

Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

查询设备0所有芯片的PCIE误码统计。

```
npu-smi info -t err-count -i 0
        NPU ID                          : 0
        Chip Count                      : 4

        Chip ID                         : 0
        Pcs Rx Error Count              : 0
        Phy Lane Error Count            : 0
        Dl Lcrc Error Count             : 0
        Dl Dcrc Error Count             : 0

        Chip ID                         : 1
        Pcs Rx Error Count              : 0
        Phy Lane Error Count            : 0
        Dl Lcrc Error Count             : 0
        Dl Dcrc Error Count             : 0

        Chip ID                         : 2
        Pcs Rx Error Count              : 0
        Phy Lane Error Count            : 0
        Dl Lcrc Error Count             : 0
        Dl Dcrc Error Count             : 0

        Chip ID                         : 3
        Pcs Rx Error Count              : 0
        Phy Lane Error Count            : 0
        Dl Lcrc Error Count             : 0
        Dl Dcrc Error Count             : 0
npu-smi info -t pcie-err -i 0
This device does not support querying pcie-err.
```

输出说明

表 10-79 打印信息说明

字段	说明
NPU ID	设备ID
Chip Count	芯片个数
Pcs Rx Error Count	pcs接收错误计数
Phy Lane Error Count	物理通道错误计数
Dl Lcrc Error Count	Dl Lcrc错误计数
Dl Dcrc Error Count	Dl DCcrc错误计数
Chip ID	芯片ID

10.1.3.4.41 查询指定芯片的 PCIE 误码计数

命令功能

npu-smi info -t err-count -i *id* **-c** *chip_id* 用于查询指定芯片的PCIE误码计数。

npu-smi info -t pcie-err -i *id* **-c** *chip_id* 用于查询指定芯片的PCIE误码计数。

命令格式

npu-smi info -t err-count -i *id* **-c** *chip_id*

npu-smi info -t pcie-err -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。 **npu-smi info -t err-count -i** *id* **-c** *chip_id*不支持指定MCU。

使用指南

npu-smi info -t err-count -i *id* **-c** *chip_id*支持的产品形态、应用场景、用户权限以及操作系统请参见表10-80。

表 10-80 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

npu-smi info -t pcie-err -i *id* **-c** *chip_id* 支持的产品形态、应用场景、用户权限以及操作系统请参见表10-81。

表 10-81 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

查询设备0中编号为0的芯片的PCIE误码计数。

```
npu-smi info -t err-count -i 0 -c 0
        Pcs Rx Error Count            : 0
        Phy Lane Error Count          : 0
        Dl Lcrc Error Count           : 0
        Dl Dcrc Error Count           : 0
npu-smi info -t pcie-err -i 0 -c 0
This device does not support querying pcie-err.
```

输出说明

表 10-82 打印信息说明

字段	说明
Pcs Rx Error Count	pcs接收错误计数
Phy Lane Error Count	物理通道错误计数
Dl Lcrc Error Count	Dl Lcrc错误计数
Dl Dcrc Error Count	Dl DCcrc错误计数

10.1.3.4.42 查询所有芯片的容器共享模式

命令功能

npu-smi info -t device-share -i *id* 用于查询所有芯片的容器共享模式。

命令格式

npu-smi info -t device-share -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-83**。

表 10-83 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

查询设备0所有芯片的容器共享模式。

```
npu-smi info -t device-share -i 0
This device does not support querying device-share.
```

输出说明

无

10.1.3.4.43 查询指定芯片的容器共享模式

命令功能

npu-smi info -t device-share -i *id* **-c** *chip_id* 用于查询指定芯片的容器共享模式。

命令格式

npu-smi info -t device-share -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-84。

表 10-84 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

查询设备0中编号为0的芯片的容器共享模式。

```
npu-smi info -t device-share -i 0 -c 0
This device does not support querying device-share.
```

输出说明

无

10.1.3.4.44 查询指定芯片的算力等级

命令功能

npu-smi info -t nve-level -i *id* **-c** *chip_id* 用于查询指定芯片的算力等级。

命令格式

npu-smi info -t nve-level -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-85**。

表 10-85 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

查询设备0中编号为0的芯片算力等级。

```
npu-smi info -t nve-level -i 0 -c 0
This device does not support querying nve-level.
```

输出说明

无

10.1.3.4.45 查询指定芯片的 AI CPU 和 Control CPU 的数量

命令功能

npu-smi info -t aicpu-config -i *id* **-c** *chip_id* 用于查询指定芯片的AI CPU和Control CPU的数量。

命令格式

npu-smi info -t aicpu-config -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-86**。

表 10-86 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

查询设备0中编号为0的芯片AI CPU和Control CPU的数量。

```
npu-smi info -t aicpu-config -i 0 -c 0
This device does not support querying aicpu-config.
```

输出说明

无

10.1.3.4.46 查询所有芯片的 AI CPU、Control CPU 和 data CPU 数量

命令功能

npu-smi info -t cpu-num-cfg -i *id* 用于查询所有芯片的AI CPU、Control CPU和data CPU数量。

命令格式

npu-smi info -t cpu-num-cfg -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

- 该命令仅支持NPU 22.0.0及以上系列版本。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-87**。

表 10-87 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

查询设备0所有芯片的AI CPU、Control CPU和data CPU数量。

```
npu-smi info -t cpu-num-cfg -i 0
This device does not support querying cpu-num-cfg.
```

输出说明

无

10.1.3.4.47 查询指定芯片的 AI CPU、Control CPU 和 data CPU 数量

命令功能

npu-smi info -t cpu-num-cfg -i *id* **-c** *chip_id* 用于查询指定芯片的AI CPU、Control CPU和data CPU数量。

命令格式

npu-smi info -t cpu-num-cfg -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

- 该命令仅支持NPU 22.0.0及以上系列版本。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-88**。

表 10-88 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

查询设备0中编号为0的芯片的AI CPU、Control CPU和data CPU数量。

```
npu-smi info -t cpu-num-cfg -i 0 -c 0
This device does not support querying cpu-num-cfg.
```

输出说明

无

10.1.3.4.48 查询指定芯片的虚拟 NPU

命令功能

npu-smi info -t info-vnpu -i *id* **-c** *chip_id* 用于查询指定芯片虚拟设备的信息。

命令格式

npu-smi info -t info-vnpu -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

- 目前算力切分功能仅应用于高校教学场景。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-89**。

表 10-89 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	NA	NA

使用实例

查询设备0中编号为0的芯片的虚拟设备的信息。

```
npu-smi info -t info-vnpu -i 0 -c 0
This device does not support querying info-vnpu.
```

输出说明

无

10.1.3.4.49 查询 NPU 设备首次上电的时间

命令功能

npu-smi info -t first-power-on-date -i *id* 用于查询NPU设备首次上电的时间。

命令格式

npu-smi info -t first-power-on-date -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

该命令仅支持NPU 21.0.4及以上系列版本。

若NPU设备首次上电时间未记录时，请24小时之后再执行此命令。

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-90**。

表 10-90 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

查询NPU设备首次上电的时间。

```
npu-smi info -t first-power-on-date -i 0
This device does not support querying first-power-on-date.
```

输出说明

无

10.1.3.4.50 查询所有芯片的进程占用内存

命令功能

npu-smi info -t proc-mem -i *id* 用于查询所有芯片的进程占用内存。

命令格式

npu-smi info -t proc-mem -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

该命令仅支持NPU 21.0.4及以上系列版本。

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-91**。

表 10-91 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0所有芯片的进程占用内存。

```
npu-smi info -t proc-mem -i 0
        NPU ID                         : 4
        Chip Count                     : 4

        Chip ID                        : 0
        Process id:26280   Process name:hlt_host_devmm_   Process memory(B):95105024
        Process id:26284   Process name:hlt_host_devmm_   Process memory(B):95109120
        Process id:26288   Process name:hlt_host_devmm_   Process memory(B):95105024

        Chip ID                        : 1
        Process id:26285   Process name:hlt_host_devmm_   Process memory(B):95105024
        Process id:26281   Process name:hlt_host_devmm_   Process memory(B):95105024

        Chip ID                        : 2
        Process id:26282   Process name:hlt_host_devmm_   Process memory(B):95105024
        Process id:26286   Process name:hlt_host_devmm_   Process memory(B):95105024

        Chip ID                        : 3
        Process id:26287   Process name:hlt_host_devmm_   Process memory(B):95105024
        Process id:26283   Process name:hlt_host_devmm_   Process memory(B):95105024
```

输出说明

表 10-92 打印信息说明

字段	说明
NPU ID	设备ID
Chip Count	芯片个数
Chip ID	芯片ID
Process id	进程ID
Process name	进程名称
Process memory(B)	进程占用的内存（单位：字节）

10.1.3.4.51 查询指定芯片的进程占用内存

命令功能

npu-smi info -t proc-mem -i *id* **-c** *chip_id*用于查询指定芯片的进程占用内存。

命令格式

npu-smi info -t proc-mem -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。 该命令不支持指定MCU。

使用指南

该命令仅支持NPU 21.0.4及以上系列版本。

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-93**。

表 10-93 部署场景

产品形态	Linux物理机	Linux容器	Windows操作系统

	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	Y	Y	NA

使用实例

查询设备0中编号为0的芯片的进程占用内存。

```
npu-smi info -t proc-mem -i 0 -c 0
        Process id:38329    Process name:hlt_host_devmm_    Process memory(B):100683776
        Process id:38321    Process name:hlt_host_devmm_    Process memory(B):100646912
```

输出说明

表 10-94 打印信息说明

字段	说明
Process id	进程ID
Process name	进程名称
Process memory(B)	进程占用的内存（单位：字节）

10.1.3.4.52 查询指定芯片异步拷贝的业务数据长度

命令功能

npu-smi info -t vnpu-svm -i *id* **-c** *chip_id* 用于查询指定芯片异步拷贝的业务数据长度。

命令格式

npu-smi info -t vnpu-svm -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

- 该命令仅支持NPU 22.0.0及以上系列版本。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-95。

表 10-95 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	NA	NA

使用实例

查询设备0中编号为0的芯片异步拷贝的业务数据长度。

```
npu-smi info -t vnpu-svm -i 0 -c 0
This device does not support querying vnpu-svm.
```

输出说明

无

10.1.3.5 配置功能（set）

10.1.3.5.1 收集日志

命令功能

npu-smi set -t collect-log -i *id*用于收集MCU的日志。收集的日志存放在/run/mcu_log目录下。

命令格式

npu-smi set -t collect-log -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

收集日志时请勿升级MCU。

系统不会自动删除日志文件，为避免日志文件占用磁盘空间，建议用户在下载后手动删除。

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-96。

表 10-96 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	N	N	NA

使用实例

收集设备0的MCU的日志。

```
npu-smi set -t collect-log -i 0
 type(0): file_len(203808)--offset(203808) [100].Logs collected successfully,(0).
 type(1): file_len(102026)--offset(102026) [100].Logs collected successfully,(1).
 type(2): file_len(101376)--offset(101376) [100].Logs collected successfully,(2).
```

输出说明

无

10.1.3.5.2 复位标卡

软件准备

带外标卡复位功能依赖ipmitool软件，需要提前下载并加载驱动。

1. 下载ipmitool软件，例如：
   - CentOS和EulerOS：**yum install ipmitool**
   - Ubuntu：**apt-get install ipmitool**
2. 加载驱动：

   modprobe ipmi_si

   modprobe ipmi_devintf

   modprobe ipmi_msghandler
3. 执行**lsmod | grep ipmi**命令查看驱动是否加载成功。如果没有加载成功，执行2重新加载驱动。

说明

如果重启OS后，需要重新执行2~3。

命令功能

npu-smi set -t reset -i *id* [**-m 1**]用于带内复位设备。

npu-smi set -t reset -i *id* 用于带外复位设备。

命令格式

npu-smi set -t reset -i *id* [**-m 1**]

npu-smi set -t reset -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

- 业务运行过程中，请不要执行此命令。
- 以上命令中[**-m 1**]表示可选参数，如果使用该参数表示通过带内复位设备，否则表示通过带外复位设备。
- Windows仅支持带内复位。
- 带外复位标卡需要依赖iBMC，为保障复位功能，请升级服务器iBMC到最新的版本。
- 设备复位成功后，请至少等待20s，再执行当前或其他设备、芯片信息查询操作，请耐心等待。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-97。

表 10-97 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	N	N	NA

使用实例

通过带外复位设备0。

```
npu-smi set -t reset -i 0
Resetting the standard PCIe card or NPU chip may cause a system hang or abnormal reset during service running.
Are you sure you want to continue resetting?(Y/N)
y
        Message                    : resetting ...
        Status                     : OK
        Message                    : Reset card successfully.
```

通过带内复位设备0。

```
npu-smi set -t reset -i 0 -m 1
Resetting the standard PCIe card or NPU chip may cause a system hang or abnormal reset during service running.
Are you sure you want to continue resetting?(Y/N)
y
        Message                    : resetting ...
        Status                     : OK
        Message                    : Reset card successfully.
```

输出说明

表 10-98 打印信息说明

字段	说明
Status	复位状态。 有如下两种状态：OK，Fail。
Message	复位提示信息

10.1.3.5.3 复位芯片

软件准备

带外芯片复位功能依赖ipmitool软件，需要提前下载并加载驱动。

1. 下载ipmitool软件，例如：
   - CentOS和EulerOS：**yum install ipmitool**
   - Ubuntu：**apt-get install ipmitool**
2. 加载驱动：

   modprobe ipmi_si

   modprobe ipmi_devintf

   modprobe ipmi_msghandler
3. 执行**lsmod | grep ipmi**命令查看驱动是否加载成功。如果没有加载成功，执行2重新加载驱动。

   说明

   如果重启OS后，需要重新执行2~3。

命令功能

npu-smi set -t reset -i *id* **-c** *chip_id* [**-m 1**]用于带内复位指定芯片。

npu-smi set -t reset -i *id* **-c** *chip_id* 用于带外复位指定芯片。

命令格式

npu-smi set -t reset -i *id* **-c** *chip_id* [**-m 1**]

npu-smi set -t reset -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。 该命令不支持指定MCU。

使用指南

- 业务运行过程中，请不要执行此命令。
- 以上命令中[**-m 1**]表示可选参数，如果使用该参数表示通过带内复位指定芯片，否则表示通过带外复位指定芯片。
- 带外复位芯片需要依赖iBMC，为保障复位功能，请升级服务器iBMC到最新的版本。
- 芯片复位成功后，请至少等待20s，再执行当前或其他芯片信息查询操作，请耐心等待。
- Windows仅支持带内复位。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-99**。

表 10-99 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	N	N	NA

使用实例

通过带外复位设备0编号为0的芯片。

```
npu-smi set -t reset -i 0 -c 0
Resetting the standard PCIe card or NPU chip may cause a system hang or abnormal reset during service running.
Are you sure you want to continue resetting?(Y/N)
y
        Message                         : resetting ...
        Status                        : OK
        Message                         : Reset chip successfully.
```

通过带内复位设备0编号为0的芯片。

```
npu-smi set -t reset -i 0 -c 0 -m 1
Resetting the standard PCIe card or NPU chip may cause a system hang or abnormal reset during service running.
Are you sure you want to continue resetting?(Y/N)
y
        Message                         : resetting ...
        Status                        : OK
        Message                         : Reset chip successfully.
```

输出说明

表 10-100 打印信息说明

字段	说明
Message	复位提示信息
Status	复位状态。 有如下两种状态：OK，Fail。

10.1.3.5.4 设置 MCU 监控状态

命令功能

npu-smi set -t mcu-monitor -i *id* **-c** *chip_id* **-d** *value* 用于设置MCU监控状态。

命令格式

npu-smi set -t mcu-monitor -i *id* **-c** *chip_id* **-d** *value*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

类型	描述
value	MCU监控模式：分为禁用、使能。 ● 0：False ● 1：True

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-101。

表 10-101 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号 3000）	N	N	N	NA

使用实例

启动设备0的MCU监控状态。

```
npu-smi set -t mcu-monitor -i 0 -c 1 -d 1
This device does not support setting mcu-monitor.
```

输出说明

无

10.1.3.5.5 设置 boot 区

命令功能

npu-smi set -t boot-area -i *id* **-c** *chip_id* 用于设置boot区。

命令格式

npu-smi set -t boot-area -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-102**。

表 10-102 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

设置设备0的boot区。

```
npu-smi set -t boot-area -i 0 -c 0
This device does not support setting boot-area.
```

输出说明

无

10.1.3.5.6 设置机械硬盘上下电

命令功能

npu-smi set -t disk-power -i *id* **-c** *chip_id* **-d** *value* 通过MCU上下电机械硬盘。

命令格式

npu-smi set -t disk-power -i *id* **-c** *chip_id* **-d** *value*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。
value	机械硬盘上下电状态。取值为0或1： ● 0：下电 ● 1：上电

使用指南

- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-103**。

表 10-103 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

通过MCU上下电机械硬盘。

```
npu-smi set -t disk-power -i 0 -c 0 -d 1
This device does not support setting disk-power.
```

输出说明

无

10.1.3.5.7 配置所有芯片的 ssh 使能状态

命令功能

npu-smi set -t ssh-enable -i *id* **-d** *value*用于配置所有芯片的ssh使能状态。

命令格式

npu-smi set -t ssh-enable -i *id* **-d** *value*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
value	ssh使能状态：分为禁用、使能。 ● 0：禁用 ● 1：使能

使用指南

- 配置所有芯片的ssh使能状态后，需要用户手动复位标卡生效配置，具体请参见**10.1.3.5.2 复位标卡**。
- 打开ssh服务之后，请务必立即修改Device的ssh登录密码（包括HwHiAiUser用户和root用户的密码）。
- 由于配置持久化功能，打开ssh服务之后不会自动关闭，建议在使用完成后及时主动关闭ssh服务。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-104**。

表 10-104 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	N	N	NA

使用实例

开启设备0中所有芯片的ssh使能。

```
npu-smi set -t ssh-enable -i 0 -d 1
[WARNING]: Enabling SSH will raise the safety risks.
      Status                      : OK
      Message                     : The ssh-enable of the card is set successfully. Reset card for the
configuration to take effect.
```

输出说明

表 10-105 打印信息说明

字段	说明
Status	命令执行结果 有如下两种状态：Fail, OK
Message	ssh使能提示信息

10.1.3.5.8 配置指定芯片的 ssh 使能状态

命令功能

npu-smi set -t ssh-enable -i *id* **-c** *chip_id* **-d** *value*用于配置指定芯片的ssh使能状态。

命令格式

npu-smi set -t ssh-enable -i *id* **-c** *chip_id* **-d** *value*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。 该命令不支持指定MCU。
value	ssh使能模式：分为禁用、使能。 ● 0：禁用 ● 1：使能

使用指南

- 配置指定芯片的ssh使能状态后，需要用户手动复位芯片生效配置，具体请参见**10.1.3.5.3 复位芯片**。
- 打开ssh服务之后，请务必立即修改Device的ssh登录密码（包括HwHiAiUser用户和root用户的密码）。
- 由于配置持久化功能，打开ssh服务之后不会自动关闭，建议在使用完成后及时主动关闭ssh服务。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-106**。

表 10-106 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	N	N	NA

使用实例

禁用设备0中编号为0的芯片的ssh使能状态。

```
npu-smi set -t ssh-enable -i 0 -c 0 -d 0
        Status                      : OK
        Message                     : The ssh-enable of the chip is set successfully. Reset chip for the
configuration to take effect.
```

输出说明

表 10-107 打印信息说明

字段	说明
Status	命令执行结果。 有如下两种状态：OK，Fail。
Message	命令执行后的提示信息。

10.1.3.5.9 配置 ECC 使能状态

命令功能

npu-smi set -t ecc-enable -i *id* **-d** *value* 用于配置所有芯片的ECC使能状态。

命令格式

npu-smi set -t ecc-enable -i *id* **-d** *value*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

类型	描述
value	ECC模式：分为禁用、使能。 ● 0：禁用 ● 1：使能

使用指南

- 配置所有芯片的ECC使能状态后，需要用户手动复位标卡生效配置，具体请参见10.1.3.5.2 复位标卡。
- ECC模式默认为禁用，若配置为使能状态后，会使某些性能下降，请谨慎开启。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-108。

表 10-108 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	N	N	NA

使用实例

开启设备0的ECC使能。

```
npu-smi set -t ecc-enable -i 0 -d 1
        Status                      : OK
        Message                     : The ecc-enable is set successfully. Reset card for the configuration to take
effect.
```

输出说明

表 10-109 打印信息说明

字段	说明
Status	命令执行结果。 有如下两种状态：OK，Fail。
Message	命令执行后的提示信息。

10.1.3.5.10 配置用户证书

命令功能

npu-smi set -t license -i *id* **-f** "*license*"用于配置用户的证书。证书只能配置一次，否则会配置失败。

命令格式

npu-smi set -t license -i *id* **-f** "*license*"

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
license	证书内容。 格式为字符串，大小为1-255个字节。 **说明** 由于Linux命令行的Shell对部分特殊字符具有保留含义，如**$$**表示Shell进程ID，**$?**表示上个命令的退出状态，故如需要设置特殊字符，用户需要对特殊字符进行转义操作。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-110**。

表 10-110 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	N	N	NA

使用实例

配置设备0的证书为abcde。

```
npu-smi set -t license -i 0 -f "abcde"
      Status                  : OK
      Message                 : The license of the card is set successfully.
```

输出说明

表 10-111 打印信息说明

字段	说明
Status	命令执行结果。 有如下两种状态：OK，Fail。
Message	命令执行后的提示信息。

10.1.3.5.11 配置用户自定义信息

命令功能

npu-smi set -t customized-info -i *id* **-f** *customized info*用于配置用户自定义信息。自定义信息只能配置一次，否则会配置失败。

命令格式

npu-smi set -t customized-info -i *id* **-f** *customized info*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
customized info	用户自定义信息。 格式为字符串，大小为1-255个字节。 **说明** 由于Linux命令行的Shell对部分特殊字符具有保留含义，如**$$**表示Shell进程ID，**$?**表示上个命令的退出状态，故如需要设置特殊字符，用户需要对特殊字符进行转义操作。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-112**。

表 10-112 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户

Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

配置设备0的用户自定义信息。

```
npu-smi set -t customized-info -i 0 -f abcde
This device does not support setting customized-info.
```

输出说明

无

10.1.3.5.12 设置 AI 芯片的 IP 信息

命令功能

npu-smi set -t ip -i *id* **-c** *chip_id* **-s** *ipstring* 用于设置AI芯片的IP信息。

命令格式

npu-smi set -t ip -i *id* **-c** *chip_id* **-s** *ipstring*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。 该命令不支持指定MCU。
ipstring	IP地址及子网掩码。 格式如下：“XX.XX.XX.XX/XX.XX.XX.XX” 其中，“/”前半部分为IP地址（IP地址的范围：192.168.1.100~192.168.1.199），后半部分为子网掩码。 **说明** IP地址的范围目前仅适用于NPU 21.0.4版本。

使用指南

- 重启系统和复位芯片后配置的IP信息都会丢失，恢复为默认IP地址。需要重新设置AI芯片的IP信息。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-113**。

表 10-113 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	N	N	NA

使用实例

对于NPU 21.0.2、NPU 21.0.3、NPU 22.0.0版本：

设置设备0，编号为0的AI芯片的IP信息为192.168.5.199，子网掩码为255.255.255.0。

```
npu-smi set -t ip -i 0 -c 0 -s 192.168.5.199/255.255.255.0
      Status                  : OK
      Message                 : The ip of the chip is set successfully.
```

对于NPU 21.0.4版本：

设置设备0，编号为0的AI芯片的IP信息为192.168.1.199，子网掩码为255.255.255.0。

```
npu-smi set -t ip -i 0 -c 0 -s 192.168.1.199/255.255.255.0
      Status                  : OK
      Message                 : The ip of the chip is set successfully.
```

输出说明

表 10-114 打印信息说明

字段	说明
Status	命令执行结果。 有如下两种状态：OK，Fail。
Message	命令执行后的提示信息。

10.1.3.5.13 清除所有芯片的 PCIE 误码计数

命令功能

npu-smi set -t errcount-clear -i *id* 用于清除所有芯片PCIE误码计数。

npu-smi set -t clear-pcie-err -i *id* 用于清除所有芯片PCIE误码计数。

命令格式

npu-smi set -t errcount-clear -i *id*

npu-smi set -t clear-pcie-err -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

npu-smi set -t errcount-clear -i *id* 支持的产品形态、应用场景、用户权限以及操作系统请参见表10-115。

表 10-115 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	N	N	NA

npu-smi set -t clear-pcie-err -i *id*支持的产品形态、应用场景、用户权限以及操作系统请参见表10-116。

表 10-116 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

清除设备0所有芯片的PCIE误码计数。

```
npu-smi set -t errcount-clear -i 0
      Status                      : OK
      Message                       : The errcount-clear  is set successfully.
```

```
npu-smi set -t clear-pcie-err -i 0
This device does not support setting clear-pcie-err.
```

输出说明

表 10-117 打印信息说明

字段	说明
Status	命令执行结果。 有如下两种状态：OK，Fail。
Message	命令执行后的提示信息。

10.1.3.5.14 清除指定芯片的 PCIE 误码计数

命令功能

npu-smi set -t errcount-clear -i *id* **-c** *chip_id* 用于清除指定芯片PCIE误码计数。

npu-smi set -t clear-pcie-err -i *id* **-c** *chip_id* 用于清除指定芯片PCIE误码计数。

命令格式

npu-smi set -t errcount-clear -i *id* **-c** *chip_id*

npu-smi set -t clear-pcie-err -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。 **npu-smi set -t errcount-clear -i** *id* **-c** *chip_id*不支持指定MCU。

使用指南

npu-smi set -t errcount-clear -i *id* **-c** *chip_id*支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-118**。

表 10-118 部署场景

产品形态	Linux物理机	Linux容器	Windows操作系统

	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	Y	N	N	NA

npu-smi set -t clear-pcie-err -i *id* **-c** *chip_id*支持的产品形态、应用场景、用户权限以及操作系统请参见表10-119。

表 10-119 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

清除设备0中编号为0的芯片的PCIE误码计数。

```
npu-smi set -t errcount-clear -i 0 -c 0
        Status                      : OK
        Message                     : The errcount-clear of the chip is set successfully.
npu-smi set -t clear-pcie-err -i 0 -c 0
This device does not support setting clear-pcie-err.
```

输出说明

表 10-120 打印信息说明

字段	说明
Status	命令执行结果。 有如下两种状态：OK，Fail。
Message	命令执行后的提示信息。

10.1.3.5.15 设置清除所有芯片 ECC 错误计数

命令功能

npu-smi set -t clear-ecc-info -i *id* 用于设置清除所有芯片ECC错误计数。

命令格式

npu-smi set -t clear-ecc-info -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

- 该命令只能清除历史累计统计及隔离页统计的ECC错误计数，ECC多bit错误计数重启设备后会自动清除。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-121**。

表 10-121 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

设置清除设备0所有芯片ECC错误计数。

```
npu-smi set -t clear-ecc-info -i 0
This device does not support setting clear-ecc-info.
```

输出说明

无

10.1.3.5.16 设置清除指定芯片 ECC 错误计数

命令功能

npu-smi set -t clear-ecc-info -i *id* **-c** *chip_id*用于设置清除指定芯片ECC错误计数。

命令格式

npu-smi set -t clear-ecc-info -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

- 该命令只能清除历史累计统计及隔离页统计的ECC错误计数，ECC多bit错误计数重启设备后会自动清除。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-122**。

表 10-122 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

设置清除设备0指定芯片ECC错误计数。

```
npu-smi set -t clear-ecc-info -i 0 -c 0
This device does not support setting clear-ecc-info.
```

输出说明

无

10.1.3.5.17 设置所有芯片的容器共享模式

命令功能

npu-smi set -t device-share -i *id* **-d** *value* 用于设置所有芯片的容器共享模式。

命令格式

npu-smi set -t device-share -i *id* **-d** *value*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
value	容器使能状态：分为禁用、使能。 ● 0：禁用 ● 1：使能

使用指南

- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-123。

表 10-123 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

设置设备0所有芯片的容器共享模式为使能。

```
npu-smi set -t device-share -i 0 -d 1
This device does not support setting device-share.
```

输出说明

无

10.1.3.5.18 设置指定芯片的容器共享模式

命令功能

npu-smi set -t device-share -i *id* **-c** *chip_id* **-d** *value* 用于设置指定芯片的容器共享模式。

命令格式

npu-smi set -t device-share -i *id* **-c** *chip_id* **-d** *value*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。
value	容器使能状态：分为禁用、使能。 ● 0：禁用 ● 1：使能

使用指南

- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-124。

表 10-124 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

设置设备0中编号为0的芯片的容器共享模式为使能。

```
npu-smi set -t device-share -i 0 -c 0 -d 1
This device does not support setting device-share.
```

输出说明

无

10.1.3.5.19 设置指定芯片的算力等级

命令功能

npu-smi set -t nve-level -i *id* **-c** ***chip_id*** **-d** *value* 用于设置指定芯片的算力等级。

命令格式

npu-smi set -t nve-level -i *id* **-c** ***chip_id*** **-d** *value*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。
value	算力等级：分为Low、Middle、High、Full。 ● 0：Low ● 1：Middle ● 2：High ● 3：Full

使用指南

- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-125**。

表 10-125 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

设置设备0中编号为0的芯片算力等级。

```
npu-smi set -t nve-level -i 0 -c 0 -d 3
This device does not support setting nve-level.
```

输出说明

无

10.1.3.5.20 设置指定芯片的 AI CPU 数量

命令功能

npu-smi set -t aicpu-config -i *id* **-c** ***chip_id*** **-d** *value* 用于设置指定芯片的AI CPU数量。

命令格式

npu-smi set -t aicpu-config -i *id* **-c** ***chip_id*** **-d** *value*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。
value	芯片的AI CPU数量。可设置为2、4、6。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-126**。

表 10-126 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

设置设备0中编号为0的芯片中AI CPU的数量为4。

```
npu-smi set -t aicpu-config -i 0 -c 0 -d 4
```

```
This device does not support setting aicpu-config.
```

输出说明

无

10.1.3.5.21 设置所有芯片的 AI CPU、Control CPU 和 data CPU 数量

命令功能

npu-smi set -t cpu-num-cfg -i ***id*** **-v** *ai_cpu_num:ctrl_cpu_num:data_cpu_num* 用于设置所有芯片的AI CPU、Control CPU和data CPU数量。

命令格式

npu-smi set -t cpu-num-cfg -i ***id*** **-v** *ai_cpu_num:ctrl_cpu_num:data_cpu_num*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
ai_cpu_num:ctrl_cpu_num:data_cpu_num	芯片的AI CPU数量：control CPU数量：data CPU数量

使用指南

- 该命令仅支持NPU 22.0.0及以上系列版本。
- 设置CPU数量后，需要复位标卡生效。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-127。

表 10-127 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号 3000）	N	N	N	NA

使用实例

设置设备0所有芯片中AI CPU、Control CPU和data CPU数量为7:1:0。

```
npu-smi set -t cpu-num-cfg -i 0 -v 7:1:0
```

```
This device does not support setting cpu-num-cfg.
```

输出说明

无

10.1.3.5.22 设置指定芯片的 AI CPU、Control CPU 和 data CPU 数量

命令功能

npu-smi set -t cpu-num-cfg -i *id* **-c** *chip_id* **-v** *ai_cpu_num:ctrl_cpu_num:data_cpu_num* 用于设置指定芯片的AI CPU、Control CPU和data CPU数量。

命令格式

npu-smi set -t cpu-num-cfg -i *id* **-c** *chip_id* **-v** *ai_cpu_num:ctrl_cpu_num:data_cpu_num*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。
ai_cpu_num:ctrl_cpu_num:data_cpu_num	芯片的AI CPU数量：control CPU数量：data CPU数量

使用指南

- 该命令仅支持NPU 22.0.0及以上系列版本。
- 设置CPU数量后，需要复位芯片生效。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-128**。

表 10-128 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号 3000）	N	N	N	NA

使用实例

设置设备0编号0的芯片中AI CPU、Control CPU和data CPU数量为7:1:0。

```
npu-smi set -t cpu-num-cfg -i 0 -c 0 -v 7:1:0
This device does not support setting cpu-num-cfg.
```

输出说明

无

10.1.3.5.23 创建指定芯片的虚拟 NPU

命令功能

NPU 22.0.0之前版本：**npu-smi set -t create-vnpu -i** *id* **-c** *chip_id* **-v** npu_number:aicore_num1:aicore_num2... 用于创建指定vnpu数量以及每个vnpu分配的aicore数量。

NPU 22.0.0及以上版本：**npu-smi set -t create-vnpu -i** *id* **-c** *chip_id* **-v** *vnpu_config* 用于创建vnpu并给vnpu分配aicore数量。

命令格式

NPU 22.0.0之前版本：**npu-smi set -t create-vnpu -i** *id* **-c** *chip_id* **-v** npu_number:aicore_num_1:aicore_num_2...：aicore_num_n

NPU 22.0.0及以上版本：**npu-smi set -t create-vnpu -i** *id* **-c** *chip_id* **-v** *vnpu_config*

参数说明

表 10-129 NPU 22.0.0 之前版本参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。
npu_number	创建的vnpu的数量，最大支持个数为16个
aicore_num_1	创建的第1个vnpu分配的aicore数量 取值1，2，4，8，16，其余为非法数据，以下参数要求一致
aicore_num_2	创建的第2个vnpu分配的aicore数量
aicore_num_n	创建的第n个vnpu分配的aicore数量，其中n等于npu_number

表 10-130 NPU 22.0.0 及以上版本参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。
vnpu_config	算力分组模板名称。

使用指南

- 目前算力切分功能仅应用于高校教学场景。
- 在芯片复位或系统重启后，已创建的虚拟NPU会自动销毁，需要重新创建。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-131**。

表 10-131 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	NA	NA

使用实例

对于NPU 22.0.0之前版本：

在设备0中编号为0的芯片上创建2个vnpu设备，每个vnpu分配4个aicore。

```
npu-smi set -t create-vnpu -i 0 -c 0 -v 2:4:4
This device does not support setting create-vnpu.
```

对于NPU 22.0.0及以上版本：

在设备0中编号为0的芯片上创建vnpu设备，此vnpu设备分配2个aicore。

```
npu-smi set -t create-vnpu -i 0 -c 0 -v vir02
This device does not support setting create-vnpu.
```

输出说明

无

10.1.3.5.24 销毁指定芯片的虚拟 NPU

命令功能

NPU 22.0.0之前版本：**npu-smi set -t destroy-vnpu -i** *id* **-c** *chip_id* **-v** vnpu_id_1:vnpu_id_2...vnpu_id_n 用于销毁指定虚拟NPU。

NPU 22.0.0及以上版本：**npu-smi set -t destroy-vnpu -i** *id* **-c** *chip_id* **-v** *vnpu_id* 用于销毁指定虚拟NPU。

命令格式

NPU 22.0.0之前版本：**npu-smi set -t destroy-vnpu -i** *id* **-c** *chip_id* -v vnpu_id_1:vnpu_id_2...vnpu_id_n

NPU 22.0.0及以上版本：**npu-smi set -t destroy-vnpu -i** *id* **-c** *chip_id* **-v** *vnpu_id*

参数说明

表 10-132 NPU 22.0.0 之前版本参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。
vnpu_id_1	销毁的第1个vnpu的id **说明** vnpu_id为65535时，表示销毁指定芯片下的所有vnpu设备
vnpu_id_2	销毁的第2个vnpu的id
vnpu_id_n	销毁的第n个vnpu的id n的范围：1~16

表 10-133 NPU 22.0.0 及以上版本参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。
vnpu_id	销毁的vnpu的id

使用指南

- 目前算力切分功能仅应用于高校教学场景。
- 在销毁指定虚拟NPU之前，请确保此设备未被使用。虚拟NPU的使用状态可参考**10.1.3.4.48 查询指定芯片的虚拟NPU**查询。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-134**。

表 10-134 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	NA	NA

使用实例

对于NPU 22.0.0之前版本：

销毁设备0编号0的芯片中编号为116、117、118的虚拟NPU设备。

```
npu-smi set -t destroy-vnpu -i 0 -c 0 -v 116:117:118
This device does not support setting destroy-vnpu.
```

对于NPU 22.0.0及以上版本：

销毁设备0编号0的芯片中编号为103的虚拟NPU设备。

```
npu-smi set -t destroy-vnpu -i 0 -c 0 -v 103
This device does not support setting destroy-vnpu.
```

输出说明

无

10.1.3.5.25 秘钥吊销

命令功能

npu-smi set -t revocate -i *id* **-c** *chip_id* **-f** *file_path* 用于实现秘钥吊销功能。

说明

- 秘钥吊销操作是不可逆的过程，吊销成功操作后，无法进行恢复，需谨慎使用。

命令格式

npu-smi set -t revocate -i *id* **-c** *chip_id* **-f** *file_path*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。
file_path	秘钥吊销文件路径。 文件路径仅支持英文字母、数字、下划线("_")、点(".")、斜线("/")、连字符("-")。 **说明** 秘钥吊销文件的获取请联系华为技术支持。

使用指南

- 此命令在确定需要进行对应的吊销操作时才可以使用，并且需要正确的吊销文件才可以吊销成功。否则，执行此命令会返回失败。
- 执行吊销操作成功后，设备将不可用。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-135。

表 10-135 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	NA	NA

使用实例

实现设备0的秘钥吊销功能。

```
npu-smi set -t revocate -i 0 -c 0 -f ***_CERT.data
This device does not support setting revocate.
```

说明

***_CERT.data请用实际的秘钥吊销文件名替换。

输出说明

无

10.1.3.5.26 设置指定芯片异步拷贝的业务数据长度

命令功能

npu-smi set -t vnpu-svm -i *id* **-c** *chip_id* **-v** *length* 用于设置指定芯片异步拷贝的业务数据长度。

命令格式

npu-smi set -t vnpu-svm -i *id* **-c** *chip_id* **-v** *length*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。
length	异步拷贝的业务数据长度。单位：Byte 取值范围为：[0，18446744073709551615]，默认值：137438953472

使用指南

- 该命令仅支持NPU 22.0.0及以上系列版本。
- 重启系统或升级后，会恢复为默认值。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-136**。

表 10-136 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	NA	NA

使用实例

设置设备0中编号为0的芯片异步拷贝的业务数据长度。

```
npu-smi set -t vnpu-svm -i 0 -c 0 -v 100
This device does not support setting vnpu-svm.
```

输出说明

无

10.1.3.6 升级功能（upgrade）

10.1.3.6.1 查询固件升级状态

命令功能

npu-smi upgrade -q *item* **-i** *id* 用于查询固件的升级状态。

命令格式

npu-smi upgrade -q *item* **-i** *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
item	固件类型。 有如下两种类型：mcu、bootloader。 该命令不支持bootloader。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-137。

表 10-137 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号 3000）	Y	N	N	NA

使用实例

查询设备1中MCU的固件升级状态。

```
npu-smi upgrade -q mcu -i 1
```

```
Conclusion                  : PASS
Message                     : The device is upgraded successfully.
```

查询设备1中bootloader的固件升级状态

```
npu-smi upgrade -q bootloader -i 1
This device does not support querying upgrade progress.
```

输出说明

表 10-138 MCU 固件升级状态打印信息说明

字段	说明
Conclusion	mcu升级结果。 有如下两种状态：PASS，Running。
Message	命令执行后的提示信息。

10.1.3.6.2 查询固件版本信息

命令功能

npu-smi upgrade -b *item* **-i** *id* 用于查询固件的版本信息。

命令格式

npu-smi upgrade -b *item* **-i** *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
item	固件类型。 有如下两种类型：mcu、bootloader。 该命令不支持bootloader。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-139**。

表 10-139 部署场景

产品形态	Linux物理机	Linux容器	Windows操作系统

	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号 3000）	Y	Y	N	NA

使用实例

查询设备0中MCU的固件版本信息。

```
npu-smi upgrade -b mcu -i 0
    Version                       : 20.1.1
```

查询设备1中bootloader的固件版本信息。

```
npu-smi upgrade -b bootloader -i 1
This device does not support querying version.
```

输出说明

表 10-140 MCU 固件版本信息打印信息说明

字段	说明
Version	mcu版本号

10.1.3.6.3 升级固件

命令功能

npu-smi upgrade -t *item* **-i** *id* **-f** *file_path*用于启动固件升级。

命令格式

npu-smi upgrade -t *item* **-i** *id* **-f** *file_path*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
item	固件类型。 有如下两种类型：mcu、bootloader。 该命令不支持bootloader。

类型	描述
file_path	升级文件路径。 文件路径仅支持英文字母、数字、下划线("_")、点(".")、斜线("/")、连字符("-")。

使用指南

- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见表10-141。

表 10-141 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号 3000）	Y	N	N	NA

使用实例

升级设备0中的MCU固件。

```
npu-smi upgrade -t mcu -i 0 -f ./IT91VPUA_MCU_V103.bin
```

```
        Validity                    : success

file_len(160847)--offset(160847) [100].
        Transfer                    : success
        Status                      : start to upgrade
Start upgrade [100].
        Status                      : OK
        Message                       : The device upgrade is started successfully
        Message                       : need active mcu
```

升级设备1中的bootloader固件。

```
npu-smi upgrade -t bootloader -i 1 -f ./bootloader.bin
This device does not support upgrading bootloader.
```

输出说明

表 10-142 升级 MCU 打印信息说明

字段	说明
Validity	mcu固件合法性校验结果

字段	说明
Transfer	mcu固件传输状态。 有如下两种状态：Fail，success。
Status	mcu固件升级状态。 有如下两种状态：Fail，OK。
Message	mcu升级命令执行后的提示信息

10.1.3.6.4 生效固件

命令功能

npu-smi upgrade -a *item* **-i** *id* 在固件升级完成之后，用该命令生效固件的新版本。

命令格式

npu-smi upgrade -a *item* **-i** *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
item	固件类型。 有如下两种类型：mcu、bootloader。 该命令不支持bootloader。

使用指南

该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-143**。

表 10-143 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号 3000）	Y	N	N	NA

使用实例

生效设备1中MCU的新版本固件。

```
npu-smi upgrade -a mcu -i 1
    Status                  : OK
    Message                 : The upgrade has taken effect after performed reboot successfully.
```

生效设备1中bootloader的新版本固件。

```
npu-smi upgrade -a bootloadetr -i 1
This device does not support activating bootloader.
```

输出说明

表 10-144 生效 MCU 打印信息说明

字段	说明
Status	mcu生效命令执行结果 有如下两种状态：Fail，OK。
Message	mcu生效命令执行后的提示信息

10.1.3.7 清除功能（clear）

10.1.3.7.1 清除所有芯片 ECC 错误计数

命令功能

npu-smi clear -t ecc-info -i *id* 用于清除所有芯片ECC错误计数。

命令格式

npu-smi clear -t ecc-info -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

- 该命令只能清除历史累计统计及隔离页统计的ECC错误计数，ECC多bit错误计数重启设备后会自动清除。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-145**。

表 10-145 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

清除设备0所有芯片ECC错误计数。

```
npu-smi clear -t ecc-info -i 0
This device does not support clearing ecc-info.
```

输出说明

无

10.1.3.7.2 清除指定芯片 ECC 错误计数

命令功能

npu-smi clear -t ecc-info -i *id* **-c** *chip_id*用于清除指定芯片ECC错误计数。

命令格式

npu-smi clear -t ecc-info -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。通过**npu-smi info -m**命令查出的Chip ID即为芯片id。

使用指南

- 该命令只能清除历史累计统计及隔离页统计的ECC错误计数，ECC多bit错误计数重启设备后会自动清除。
- 该命令支持的产品形态、应用场景、用户权限以及操作系统请参见**表10-146**。

表 10-146 部署场景

产品形态	Linux物理机		Linux容器	Windows操作系统
	root用户	运行用户组（非root用户）	root用户	管理员用户
Atlas 300I 推理卡（型号：3000）	N	N	N	NA

使用实例

清除设备0指定芯片ECC错误计数。

```
npu-smi clear -t ecc-info -i 0 -c 0
This device does not support clearing ecc-info.
```

输出说明

无

10.1.4 npu-smi 命令介绍（适用于 1.0.0-1.0.10 版本）

10.1.4.1 简介

npu-smi是npu的系统管理工具，可以收集Atlas 300I 推理卡或Ascend 310的信息，查看显存使用情况，并具有启用和禁用npu配置选项，执行固件升级等功能。使用NPU 20.X.X系列版本软件包安装，虚拟机支持下发部分npu-smi命令。

10.1.4.2 npu-smi 使用方法

命令通用格式：**npu-smi** *Command Options...*

参数说明

表 10-147 Command 参数说明

参数值	描述
info	查询信息
set	配置芯片占用模式、显存和复位功能
upgrade	升级功能配置和查询接口
-h	显示命令提示信息
-v	显示npu-smi工具的版本

说明

Options 参数的详细提示信息可通过执行**npu-smi** *command*命令获取。

10.1.4.3 查询帮助

命令功能

npu-smi -h | --help 用于显示工具的帮助信息。

命令格式

npu-smi -h

使用实例

#显示npu-smi工具的帮助信息。

- 对于NPU 20.2.X及以上系列版本，如下：

```
npu-smi -h
Usage: npu-smi <Command|-h|-v> [Options...]

npu-smi Command:
        -h          This command’s help information
        --help       This command’s help information
        -v          Show npu-smi version information
        info         Show hardware details
        set         Modify device configuration properties
        upgrade      Upgrade card or MCU firmware
```

- 对于NPU 1.X.X、NPU 20.0.X以及NPU 20.1.X系列版本，如下：

```
npu-smi -h
Usage: npu-smi <Command|-h|-v> [Options...]

npu-smi Command:
        -h          This help or command's help
        --help       This help or command's help
        -v          Show npu-smi version information
        info         Show hardware details
        set         Modify device configuration properties
        upgrade      Upgrade card or MCU firmware
```

10.1.4.4 查询 npu-smi 工具版本

命令功能

npu-smi -v用于查询npu-smi工具的版本信息。

命令格式

npu-smi -v

使用实例

#查询npu-smi的版本信息。

```
npu-smi -v
npu-smi Version                 : 1.0.2
```

10.1.4.5 信息查询（info）

10.1.4.5.1 查询基本信息

命令功能

npu-smi info 用于查询所有设备的基本信息。

命令格式

npu-smi info

使用指南

无。

使用实例

#查询所有设备的基本信息。

```
npu-smi info
+------------------------------------------------------------------------------+
| npu-smi 20.2.0                       Version: 20.2.0                         |
+-------------------+-----------------+--------------------------------------+
| NPU     Name      | Health          | Power(W)          Temp(C)            |
| Chip    Device    | Bus-Id          | AICore(%)         Memory-Usage(MB)   |
+===================+=================+======================================+
| 215     310       | OK              | 12.8              56                 |
| 0       0         | 0000:DA:00.0    | 0                 2703 / 8192        |
+-------------------+-----------------+--------------------------------------+
| 215     310       | OK              | 12.8              57                 |
| 1       1         | 0000:DC:00.0    | 0                 2703 / 8192        |
+-------------------+-----------------+--------------------------------------+
| 215     310       | OK              | 12.8              57                 |
| 2       2         | 0000:DD:00.0    | 0                 2703 / 8192        |
+-------------------+-----------------+--------------------------------------+
| 215     310       | OK              | 12.8              55                 |
| 3       3         | 0000:DB:00.0    | 0                 2703 / 8192        |
+===================+=================+======================================+
```

输出说明

表 10-148 打印信息说明

字段	说明
npu-smi	npu-smi 工具版本。
Version	驱动版本。
NPU	设备ID。
Name	芯片名称。
Health	芯片的健康状态。 可取值为：OK、Warning、Alarm、Critical或UNKNOWN。

字段	说明
Power(W)	芯片功率。
Temp(C)	芯片温度。
Chip	芯片ID
Device	芯片编号。
Bus-Id	BUS ID。
AICore(%)	ai core占用率。
Memory-Usage(MB)	内存占比。

10.1.4.5.2 查询芯片监控数据

命令功能

npu-smi info watch -i *id* **-c** *chip_id* **-d** *delay_seconds* **-s** *watch_type* 用于查询所有或单个芯片的监控数据。

命令格式

npu-smi info watch -i *id* **-c** *chip_id* **-d** *delay_seconds* **-s** *watch_type*

使用指南

无。

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。如果有4个芯片，则编号为0、1、2、3。
delay_seconds	每轮查询延迟时长，单位为秒。
watch_type	监控类型。p代表功率，t代表温度，a代表ai core占用率，i代表ai cpu占用率，c代表control cpu占用率，m代表内存占用率，b代表内存带宽占用率。

使用实例

#查询设备199编号为2的芯片的监控数据。

```
npu-smi info watch -i 199 -c 2 -d 5 -s ptaicmb
Npu ID    Chip Id   Pwr       Temp      AI Core   AI Cpu    Ctrl Cpu  Memory    Memory BW
Idx       Idx       W         C         %         %         %         %         %
```

```
199     2     12.8    54     0     0     0     29     0
199     2     12.8    54     0     0     0     29     0
199     2     12.8    54     0     0     0     29     0
199     2     12.8    54     0     0     0     29     0
199     2     12.8    54     0     0     0     29     0
```

10.1.4.5.3 查询所有 NPU 设备

命令功能

npu-smi info -l 用于查询所有NPU设备。

命令格式

npu-smi info -l

使用指南

虚拟机和docker容器中不支持查询product Name，Serial Number域。

使用实例

#查询所有NPU设备的相关信息。

- 对于NPU 20.2.X及以上系列版本，如下：

```
npu-smi info -l
Card Count                    : 2

NPU ID                        : 0
Product Name                  : IT21DMPA01
Serial Number                 : 032WHW10K0000818
Chip Count                    : 4

NPU ID                        : 1
Product Name                  : IT21DMPC04
Serial Number                 : 034BDJ10M3000020
Chip Count                    : 4
```

- 对于NPU 1.X.X、NPU 20.0.X以及NPU 20.1.X系列版本，如下：

```
npu-smi info -l
Total Count                   : 2

NPU ID                        : 0
Product Name                  : IT21DMPB01
Serial Number                 : 033EFS10L6001937
Chip Count                    : 4

NPU ID                        : 1
Product Name                  : IT21DMPB01
Serial Number                 : 033EFS10L6002408
Chip Count                    : 4
```

10.1.4.5.4 查询所有芯片映射关系信息

命令功能

npu-smi info -m 用于查询所有芯片的映射关系信息。

命令格式

npu-smi info -m

使用指南

虚拟机和docker容器中不显示MCU信息。

使用实例

#查询目前所有芯片的映射关系信息。

```
npu-smi info -m
NPU ID          Chip ID          Chip Logic ID      Chip Name
18              0                0                  MiniD
35              0                1                  MiniD
```

10.1.4.5.5 查询具体卡信息

命令功能

npu-smi info -t board -i *id* 用于查询具体卡的board信息。

命令格式

npu-smi info -t board -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

- 虚拟机和docker容器中不支持查询具体卡的board信息。
- 该命令仅支持在root用户下执行。

使用实例

#查询设备1的具体信息。

- 对于NPU 20.2.X及以上系列版本，如下：

```
npu-smi info -t board -i 1
NPU ID                      : 1
Product Name                : IT21DMPA01
Model                       : NA
Manufacturer                : Huawei
Serial Number               : 032WHW10K0000818
Software Version            : 20.2.0
Firmware Version            : 1.76.22.3.220
Board ID                    : 0xfc
PCB ID                      : D
BOM ID                      : 1
PCIe Bus Info               : 0000:86:00.0
Slot ID                     : 4
Class ID                    : NA
PCI Vendor ID               : 0x19E5
PCI Device ID               : 0xD100
Subsystem Vendor ID         : 0x0200
```

```
Subsystem Device ID            : 0x0100
Chip Count                     : 4
Faulty Chip Count              : 0
```

- 对于NPU 1.X.X、NPU 20.0.X以及NPU 20.1.X系列版本，如下：

```
npu-smi info -t board -i 1
NPU ID                         : 1
Product Name                   : NPU
Model                          : Hi***
Manufacturer                   : Huawei
Serial Number                  : RFD1644N48373
Software Version               : 1.1.0T.B600
Firmware Version               : 1.1.0.600
Board ID                       : 0xfc
PCB ID                         : C
BOM ID                         : 0
PCIe Bus Info                  : 0000:2d:00.0
Slot ID                        : 1
Class ID                       : NA
PCI Vendor ID                  : 0x19E5
PCI Device ID                  : 0xD100
Subsystem Vendor ID            : 0x0200
Subsystem Device ID            : 0x0100
Chip Count                     : 4
Chip fault                     : 0
```

10.1.4.5.6 查询所有芯片常用信息

命令功能

npu-smi info -t common -i *id* 用于查询所有芯片常用信息。

命令格式

npu-smi info -t common -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

虚拟机和docker容器中不支持该命令。

使用实例

#查询设备1中所有芯片的常用信息。

```
npu-smi info -t common -i 1
NPU ID                         : 1
Chip Count                     : 4

Chip ID                        : 0
Memory Usage Rate(%)           : 30
Aicore Usage Rate(%)           : 0
Temperature(C)                 : 46
Rated Power Dissipation(W)     : 12.8
```

```
Chip ID                         : 1
Memory Usage Rate(%)            : 30
Aicore Usage Rate(%)            : 0
Temperature(C)                  : 47
Rated Power Dissipation(W)      : 12.8

Chip ID                         : 2
Memory Usage Rate(%)            : 30
Aicore Usage Rate(%)            : 0
Temperature(C)                  : 48
Rated Power Dissipation(W)      : 12.8

Chip ID                         : 3
Memory Usage Rate(%)            : 30
Aicore Usage Rate(%)            : 0
Temperature(C)                  : 48
Rated Power Dissipation(W)      : 12.8

Chip Name                       : MCU
Temperature(C)                  : 45
Power Dissipation(W)            : 19.6
```

说明

以上回显中，固件版本为1.73.5.5及之下Chip Name显示为Chip ID。

10.1.4.5.7 查询所有芯片闪存信息

命令功能

npu-smi info -t flash -i *id* 用于查询所有芯片的闪存信息。

命令格式

npu-smi info -t flash -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

无。

使用实例

#查询设备1中所有芯片的闪存信息。

```
npu-smi info -t flash -i 1
npu id                          : 1
chip count                      : 4

chip id                         : 0
flash count                     : 1
```

```
flash id                    : 624427202
manufacturer id               : 0xc2
capacity(mb)                  : 16

chip id                     : 1
flash count                   : 1

flash id                    : 624427202
manufacturer id               : 0xc2
capacity(mb)                  : 16

chip id                     : 2
flash count                   : 1

flash id                    : 624427202
manufacturer id               : 0xc2
capacity(mb)                  : 16

chip id                     : 3
flash count                   : 1

flash id                    : 624427202
manufacturer id               : 0xc2
capacity(mb)                  : 16
```

10.1.4.5.8 查询所有芯片内存信息

命令功能

npu-smi info -t memory -i *id* 用于查询所有芯片的内存信息。

命令格式

npu-smi info -t memory -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

无。

使用实例

#查询设备1的所有芯片的内存信息。

```
npu-smi info -t memory -i 1
NPU ID                      : 1
Chip Count                    : 4

Chip ID                     : 0
Capacity(MB)                  : 8192
Clock Speed(MHz)                : 400

Chip ID                     : 1
```

```
Capacity(MB)                : 8192
Clock Speed(MHz)            : 400

Chip ID                     : 2
Capacity(MB)                : 8192
Clock Speed(MHz)            : 400

Chip ID                     : 3
Capacity(MB)                : 8192
Clock Speed(MHz)            : 400
```

10.1.4.5.9 查询所有芯片统计信息

命令功能

npu-smi info -t usages -i *id* 用于统计所有芯片的利用状况信息。

命令格式

npu-smi info -t usages -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

无。

使用实例

#查询设备1所有芯片内存等利用状况信息。

```
npu-smi info -t usages -i 1
NPU ID                      : 1
Chip Count                  : 4

Chip ID                     : 0
Memory Capacity(MB)         : 8192
Memory Usage Rate(%)        : 38
Aicore Usage Rate(%)        : 50
Aicpu Usage Rate(%)         : 3
Ctrlcpu Usage Rate(%)       : 0
Memory Bandwidth Usage Rate(%) : 13

Chip ID                     : 1
Memory Capacity(MB)         : 8192
Memory Usage Rate(%)        : 40
Aicore Usage Rate(%)        : 0
Aicpu Usage Rate(%)         : 1
Ctrlcpu Usage Rate(%)       : 0
Memory Bandwidth Usage Rate(%) : 4

Chip ID                     : 2
Memory Capacity(MB)         : 8192
Memory Usage Rate(%)        : 39
```

```
Aicore Usage Rate(%)          : 0
Aicpu Usage Rate(%)           : 1
Ctrlcpu Usage Rate(%)         : 0
Memory Bandwidth Usage Rate(%) : 3

Chip ID                       : 3
Memory Capacity(MB)           : 8192
Memory Usage Rate(%)          : 39
Aicore Usage Rate(%)          : 55
Aicpu Usage Rate(%)           : 0
Ctrlcpu Usage Rate(%)         : 0
Memory Bandwidth Usage Rate(%) : 13
```

10.1.4.5.10 查询所有芯片温度

命令功能

npu-smi info -t temp -i *id* 用于查询所有芯片的温度。

命令格式

npu-smi info -t temp -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

虚拟机和docker容器中不显示MCU信息。

使用实例

#查询设备1的所有芯片的温度。

```
npu-smi info -t temp -i 1
NPU ID                        : 1
Chip Count                    : 4

Chip ID                       : 0
Temperature(C)                : 43

Chip ID                       : 1
Temperature(C)                : 46

Chip ID                       : 2
Temperature(C)                : 45

Chip ID                       : 3
Temperature(C)                : 45

Temperature (C)               : 39
Chip Name                     : MCU
```

说明

以上回显中，固件版本为1.73.5.5及之下Chip Name显示为Chip ID。

10.1.4.5.11 查询所有芯片功率

命令功能

npu-smi info -t power -i *id* 用于查询所有芯片的功率。

命令格式

npu-smi info -t power -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

虚拟机和docker容器中不显示MCU信息。

使用实例

#查询设备1的所有芯片的功率。

```
npu-smi info -t power -i 1
NPU ID                      : 1
Chip Count                  : 4

Chip ID                     : 0
Rated Power Dissipation(W)     : 12.8

Chip ID                     : 1
Rated Power Dissipation(W)     : 12.8

Chip ID                     : 2
Rated Power Dissipation(W)     : 12.8

Chip ID                     : 3
Rated Power Dissipation(W)     : 12.8

Power Dissipation(W)          : 21.9
Chip Name                   : MCU
```

说明

以上回显中，固件版本为1.73.5.5及之下Chip Name显示为Chip ID。

10.1.4.5.12 查询所有芯片电压

命令功能

npu-smi info -t volt -i *id* 用于查询所有芯片的电压。

命令格式

npu-smi info -t volt -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

虚拟机和docker容器中不显示MCU信息。

使用实例

#查询设备1的所有芯片电压信息。

```
npu-smi info -t volt -i 1
NPU ID                    : 1
Chip Count                : 4

Chip ID                   : 0
Voltage (V)               : 0.80

Chip ID                   : 1
Voltage (V)               : 0.80

Chip ID                   : 2
Voltage (V)               : 0.80

Chip ID                   : 3
Voltage (V)               : 0.80

Voltage (V)               : 11.88
Chip Name                 : MCU
```

说明

以上回显中，固件版本为1.73.5.5及之下Chip Name显示为Chip ID。

10.1.4.5.13 查询所有芯片健康状态

命令功能

npu-smi info -t health -i *id* 用于查询所有芯片的健康状态。

命令格式

npu-smi info -t health -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

虚拟机和docker容器中不显示MCU状态。

使用实例

#查询设备1所有芯片的健康状态。

```
npu-smi info -t health -i 1
NPU ID                    : 1
Chip Count                : 4

Chip ID                   : 0
Health                    : OK

Chip ID                   : 1
Health                    : OK

Chip ID                   : 2
Health                    : OK

Chip ID                   : 3
Health                    : OK

Health                    : OK
Chip Name                 : MCU
```

说明

以上回显中，固件版本为1.73.5.5及之下Chip Name显示为Chip ID。

输出说明

表 10-149 打印信息说明

字段	说明
NPU ID	设备ID。
Chip Count	芯片数量。
Chip ID	芯片ID。
Health	芯片的健康状态。 可取值为：OK、Warning、Alarm、Critical或UNKNOWN。
Chip Name	芯片名称。

10.1.4.5.14 查询所有芯片的 ssh 使能状态

命令功能

npu-smi info -t ssh-enable -i *id* 用于查询所有芯片的ssh使能状态。

命令格式

npu-smi info -t ssh-enable -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

无。

使用实例

#查询设备1所有芯片的ssh使能状态。

```
npu-smi info -t ssh-enable -i 1
NPU ID                    : 1
Chip Count                : 4

Chip ID                   : 0
SSH-Enable Status         : False

Chip ID                   : 1
SSH-Enable Status         : False

Chip ID                   : 2
SSH-Enable Status         : False

Chip ID                   : 3
SSH-Enable Status         : False
```

10.1.4.5.15 查询所有芯片 ECC 错误计数

命令功能

npu-smi info -t ecc -i *id* 用于查询所有芯片的ECC错误计数。

命令格式

npu-smi info -t ecc -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

无。

使用实例

#查询设备1所有芯片的ECC错误统计。

- 对于NPU 20.2.X及以上系列版本，如下：

```
npu-smi info -t ecc -i 1
NPU ID                    : 1
Chip Count                : 4

Chip ID                   : 0
Single-bit Error Count    : 0
Double-bit Error Count    : 0

Chip ID                   : 1
Single-bit Error Count    : 0
Double-bit Error Count    : 0


Chip ID                   : 2
Single-bit Error Count    : 0
Double-bit Error Count    : 0


Chip ID                   : 3
Single-bit Error Count    : 0
Double-bit Error Count    : 0
```

- 对于NPU 1.X.X、NPU 20.0.X以及NPU 20.1.X系列版本，如下：

```
npu-smi info -t ecc -i 1
NPU ID                    : 1
Chip Count                : 4

Chip ID                   : 0
Single Bit Error Count    : 0
Double Bit Error Count    : 0

Chip ID                   : 1
Single Bit Error Count    : 0
Double Bit Error Count    : 0


Chip ID                   : 2
Single Bit Error Count    : 0
Double Bit Error Count    : 0


Chip ID                   : 3
Single Bit Error Count    : 0
Double Bit Error Count    : 0
```

10.1.4.5.16 查询 ECC 使能状态

命令功能

npu-smi info -t ecc-enable -i *id* 用于查询PCIe卡的ECC使能状态。

命令格式

npu-smi info -t ecc-enable -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

无。

使用实例

#查询设备1的ECC使能状态。

```
npu-smi info -t ecc-enable -i 1
ECC-Enable Status                   : True
```

10.1.4.5.17 查询 p2p 使能状态

命令功能

npu-smi info -t p2p-enable -i *id* 用于查询PCIe卡的p2p使能状态。

命令格式

npu-smi info -t p2p-enable -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info --l**命令查出的NPU ID即为设备id。

使用指南

无。

使用实例

#查询设备1的p2p使能状态。

```
npu-smi info -t p2p-enable -i 1
p2p-Enable Status                   : False
```

10.1.4.5.18 查询指定芯片信息

命令功能

npu-smi info -t board -i *id* **-c** *chip_id* 用于查询PCIe卡上某一芯片的board信息。

命令格式

npu-smi info -t board -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。如果有4个芯片，则编号为0、1、2、3。

使用指南

该命令仅支持在root用户下执行。

使用实例

#查询设备1中编号为1的芯片的board信息。

- 对于NPU 21.0.X系列版本，如下：

```
npu-smi info -t board -i 1 -c 1
NPU ID                          : 1
Chip ID                         : 1
Chip Type                       : Ascend
Chip Name                       : 310
Board ID                        : 0xc8
PCB ID                          : NA
BOM ID                          : 0
Die ID                          : V100, 088F5D94 02410E10 A5127993 0100A8C0 26100063
SLOT ID                         : 2
PCIe Bus Info                   : 0000:8A:00.0
firmware Version                : 1.77.22.3.220
```

- 对于NPU 1.X.X、NPU 20.0.X、NPU 20.1.X以及NPU 20.2.X系列版本，如下：

```
npu-smi info -t board -i 1 -c 1
NPU ID                          : 1
Chip ID                         : 1
Chip Type                       : HIXXXX
Chip Name                       : HI1910
Board ID                        : 0x01
PCB ID                          : @
BOM ID                          : 0
Die ID                          : V100, 918F2194 2161A900 AC443652 0100A8C0 8E100063
SLOT ID                         : 2
PCIe Bus Info                   : 0000:33:00.0
firmware Version                : 1.1.0.600
```

10.1.4.5.19 查询指定芯片温度

命令功能

npu-smi info -t temp -i *id* **-c** *chip_id* 用于查询某个芯片的温度。

命令格式

npu-smi info -t temp -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。如果有4个芯片，则编号为0、1、2、3。

使用指南

无。

使用实例

#查询设备1中编号为1的芯片的温度。

```
npu-smi info -t temp -i 1 -c 1
Temperature (C)                 : 49
```

10.1.4.5.20 查询指定芯片功率

命令功能

npu-smi info -t power -i *id* **-c** *chip_id* 用于查询某个芯片的功率。

命令格式

npu-smi info -t power -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。如果有4个芯片，则编号为0、1、2、3。

使用指南

无。

使用实例

#查询设备1中编号为1的芯片功率。

```
npu-smi info -t power -i 1 -c 1
Rated Power Dissipation(W)      : 12.8
```

10.1.4.5.21 查询指定芯片电压

命令功能

npu-smi info -t volt -i *id* **-c** *chip_id* 用于查询某个芯片的电压。

命令格式

npu-smi info -t volt -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info --l**命令查出的NPU ID即为设备id。
chip_id	芯片id。如果有4个芯片，则编号为0、1、2、3。

使用说明

无。

使用实例

#查询设备1中编号为1的芯片的电压。

```
npu-smi info -t volt -i 1 -c 1
Voltage(V)                    : 0.80
```

10.1.4.5.22 查询指定芯片统计信息

命令功能

npu-smi info -t usages -i *id* **-c** *chip_id* 用于查询某个芯片的统计信息。

命令格式

npu-smi info -t usages -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。如果有4个芯片，则编号为0、1、2、3。

使用指南

无。

使用实例

#查询设备1中编号为1的芯片的统计信息。

```
npu-smi info -t usages -i 1 -c 1
Memory Capacity(MB)            : 8192
Memory Usage Rate(%)           : 30
Aicore Usage Rate(%)           : 5
Aicpu Usage Rate(%)            : 1
Ctrlcpu Usage Rate(%)          : 0
Memory Bandwidth Usage Rate(%) : 12
```

10.1.4.5.23 查询指定芯片闪存信息

命令功能

npu-smi info -t flash -i *id* **-c** *chip_id* 用于查询某个芯片的闪存信息。

命令格式

npu-smi info -t flash -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。如果有4个芯片，则编号为0、1、2、3。

使用指南

无。

使用实例

#查询设备1中编号为1的芯片的闪存信息。

```
npu-smi info -t flash -i 1 -c 1
Flash Count                    : 1

Flash ID                       : 624427202
Manufacturer ID                : 0xC2
Capacity(MB)                   : 16
```

10.1.4.5.24 查询指定芯片内存信息

命令功能

npu-smi info -t memory -i *id* **-c** *chip_id* 用于查询某个芯片的内存信息。

命令格式

npu-smi info -t memory -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。如果有4个芯片，则编号为0、1、2、3。

使用指南

无。

使用实例

#查询设备1中编号为1的芯片的内存信息。

```
npu-smi info -t memory -i 1 -c 1
Capacity(MB)                    : 8192
Clock Speed(MHz)                : 1600
```

10.1.4.5.25 查询指定芯片健康状态

命令功能

npu-smi info -t health -i *id* **-c** *chip_id* 用于查询某个芯片的健康状态。

命令格式

npu-smi info -t health -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。如果有4个芯片，则编号为0、1、2、3。

使用指南

无。

使用实例

#查询设备1中编号为1的芯片的健康状态。

```
npu-smi info -t health -i 1 -c 1
Health Status                  : OK
Error Code                     : NA
Error Information              : NA
```

若出现如下回显中有错误码信息，错误码信息详情请参见《Atlas 300I 推理卡 带外管理接口规范（型号 3000, 3010）》中“故障定义”章节。

```
Health Status                  : Alarm
Error Code                     : 1D5B 1D51 1D55 1D58 1D60 1C1F 1C1C 1D8F 1D4C
Error Information              :
                               : 7515
                               : 7505
                               : 7509
                               : 7512
                               : 7520
                               : 7199
                               : 7196
                               : 7567
                               : 7500
```

输出说明

表 10-150 打印信息说明

字段	说明
Health Status	芯片的健康状态。 可取值为：OK、Warning、Alarm、Critical或UNKNOWN。
Error Code	错误码。
Error Information	错误信息。

10.1.4.5.26 查询指定芯片的 ssh 使能状态

命令功能

npu-smi info -t ssh-enable -i *id* **-c** *chip_id* 用于查询某个芯片的ssh使能状态。

命令格式

npu-smi info -t ssh-enable -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。如果有4个芯片，则编号为0、1、2、3。

使用指南

无。

使用实例

#查询设备1中编号为0的芯片的ssh使能状态。

```
npu-smi info -t ssh-enable -i 1 -c 0
SSH-Enable Status              : False
```

10.1.4.5.27 查询指定芯片 ECC 错误计数

命令功能

npu-smi info -t ecc -i *id* **-c** *chip_id* 用于查询某个芯片的ECC错误计数。

命令格式

npu-smi info -t ecc -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。如果有4个芯片，则编号为0、1、2、3。

使用指南

无。

使用实例

#查询设备1中编号为1的芯片的ECC错误计数。

- 对于NPU 20.2.X及以上系列版本，如下：
```
npu-smi info -t ecc -i 1 -c 1
Single-bit Error Count         : 25
Double-Bit Error Count         : 0
```
- 对于NPU 1.X.X、NPU 20.0.X以及NPU 20.1.X系列版本，如下：
```
npu-smi info -t ecc -i 1 -c 1
Single Bit Error Count         : 25
Double Bit Error Count         : 0
```

10.1.4.5.28 查询 NPU 与 MCU 之间的 IIC 通道是否正常

命令功能

npu-smi info -t i2c_check -i *id* 用于查询NPU与MCU之间的IIC通道是否正常。

命令格式

npu-smi info -t i2c_check -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

虚拟机和docker容器中不支持该命令。

使用实例

#查询设备1与MCU之间的IIC通道状态。

```
npu-smi info -t i2c_check -i 1
Checking... Please do not interrupt!
miniD 1 to MCU iic channel status: OK
miniD 3 to MCU iic channel status: OK
MCU to miniD 0 iic channel status: OK
MCU to miniD 1 iic channel status: OK
MCU to miniD 2 iic channel status: OK
MCU to miniD 3 iic channel status: OK
```

10.1.4.5.29 查询指定芯片的产品类型

命令功能

npu-smi info -t product -i *id* **-c** *chip_id* 用于查询指定AI芯片的产品类型。

命令格式

npu-smi info -t product -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。 通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。 如果有4个芯片，则编号为0、1、2、3。

使用指南

- 对于NPU 1.X.X系列版本：NPU 1.4.0及以上版本支持此命令。

- 对于NPU 20.X.X及以上系列版本：所有版本均支持此命令。

使用实例

#查询设备0，编号为2的AI芯片的产品类型。具体产品型号以实际查询的产品为准。

- 对于NPU 20.2.X及以上系列版本，如下：
```
npu-smi info -t product -i 0 -c 2
product type:  Atlas 300I Model 3000.
```
- 对于NPU 1.X.X、NPU 20.0.X以及NPU 20.1.X系列版本，如下：
```
npu-smi info -t product -i 0 -c 2
product type:  Atlas 300 Model 3000.
```

10.1.4.5.30 查询指定芯片的 Matrix 进程监控数据

命令功能

npu-smi info proc -i *id* **-c** *chip_id* **-d** delay_seconds 用于查询指定芯片的Matrix进程监控数据。

命令格式

npu-smi info proc -i *id* **-c** *chip_id* **-d** delay_seconds

参数说明

类型	描述
id	设备id。 通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。 如果有4个芯片，则编号为0、1、2、3。
delay_seconds	每轮查询延迟时长，单位为秒。默认为1秒，取值范围：1~100。

使用指南

该命令只能在Atlas 300I 推理卡NPU版本为1.4.0情况下使用。

使用实例

#查询设备0，编号为2的芯片在2s内的Matrix进程监控数据。

```
npu-smi info proc -i 0 -c 2 -d 2
Npu ID    Chip Id   PID       Memory%  AI Cpu%
0         2         0         0.0      0.0
0         2         1         0.0      0.0
0         2         2         0.0      0.0
0         2         3         100.0    100.0
0         2         4         100.0    100.0
0         2         5         100.0    100.0
0         2         6         38.3     88.6
```

```
0       2       7       77.7    91.5
0       2       8       79.3    33.5
```

10.1.4.5.31 查询用户证书

命令功能

npu-smi info -t license -i *id* 用于查询用户的证书。

命令格式

npu-smi info -t license -i *id*

参数说明

类型	描述
id	设备id。 通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

- 对于NPU 1.X.X系列版本：该命令仅支持npu-smi及MCU版本为1.4.0及以上版本。
- 对于NPU 20.X.X及以上系列版本：所有版本均支持此命令。
- 虚拟机和docker容器中不支持该命令。

使用实例

#获取设备1的证书为abcde。

```
npu-smi info -t license -i 1
License             : abcde
```

10.1.4.5.32 查询 AI 芯片的 IP 信息

命令功能

npu-smi info -t ip -i *id* **-c** *chip_id* 用于查询AI芯片的IP信息。

命令格式

npu-smi info -t ip -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。 通过**npu-smi info -l**命令查出的NPU ID即为设备id。

类型	描述
chip_id	芯片id。 如果有4个芯片，则编号为0、1、2、3。

使用指南

- 对于NPU 1.X.X系列版本：NPU 1.4.1及以上版本支持此命令。
- 对于NPU 20.X.X及以上系列版本：所有版本均支持此命令。

使用实例

#查询设备0中编号为0的AI芯片的IP信息。

```
npu-smi info -t ip -i 0 -c 0
ip                          : 192.168.4.199
netmask                     : 255.255.255.0
```

10.1.4.5.33 查询所有芯片的 PCIE 误码计数

命令功能

npu-smi info -t err-count -i *id* 用于查询所有芯片PCIE误码计数。

命令格式

npu-smi info -t err-count -i *id*

参数说明

类型	描述
id	设备id。 通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

- 该命令仅支持NPU 20.1.X及以上系列版本。

使用实例

#查询设备1所有芯片的PCIE误码统计。

```
npu-smi info -t err-count -i 1
NPU ID                      : 1
Chip Count                  : 4

Chip ID                     : 0
Pcs Rx Error Count          : 25
Phy Lane Error Count        : 0
Dl Lcrc Error Count         : 25
Dl Dcrc Error Count         : 0
```

```
Chip ID                    : 1
Pcs Rx Error Count           : 25
Phy Lane Error Count          : 0
Dl Lcrc Error Count          : 25
Dl Dcrc Error Count           : 0

Chip ID                    : 2
Pcs Rx Error Count           : 25
Phy Lane Error Count          : 0
Dl Lcrc Error Count          : 25
Dl Dcrc Error Count           : 0

Chip ID                    : 3
Pcs Rx Error Count           : 25
Phy Lane Error Count          : 0
Dl Lcrc Error Count          : 25
Dl Dcrc Error Count           : 0
```

10.1.4.5.34 查询指定芯片的 PCIE 误码计数

命令功能

npu-smi info -t err-count -i *id* **-c** *chip_id* 用于查询指定芯片的PCIE误码计数。

命令格式

npu-smi info -t err-count -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。如果有4个芯片，则编号为0、1、2、3。

使用指南

- 该命令仅支持NPU 20.1.X及以上系列版本。

使用实例

#查询设备1中编号为1的芯片的PCIE误码计数。

```
npu-smi info -t err-count -i 1 -c 1
Pcs Rx Error Count           : 25
Phy Lane Error Count          : 0
Dl Lcrc Error Count          : 25
Dl Dcrc Error Count           : 0
```

10.1.4.6 配置功能（set）

10.1.4.6.1 配置所有芯片的 ssh 使能状态

命令功能

npu-smi set -t ssh-enable -i *id* **-d** *value*用于配置所有芯片的ssh使能状态。

命令格式

npu-smi set -t ssh-enable -i *id* **-d** *value*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
value	ssh使能状态：分为禁用、使能。 ● 0：禁用 ● 1：使能

使用指南

- 虚拟机和docker容器中不支持该命令。
- 该命令仅支持在root用户下执行。
- 配置所有芯片的ssh使能状态后，需要用户手动复位标卡生效配置，具体请参见**10.1.4.6.6 复位标卡**。
- 打开ssh服务之后，请务必立即修改Device的ssh登录密码（包括HwHiAiUser用户和root用户的密码）。
- 由于配置持久化功能，打开ssh服务之后不会自动关闭，建议在使用完成后及时主动关闭ssh服务。

使用实例

#开启设备1中所有芯片的ssh使能。

- 对于NPU 20.2.X及以上系列版本，如下：
```
npu-smi set -t ssh-enable -i 1 -d 1
Status                  : OK
Message                 : The ssh-enable of the card is set successfully. Reset card for the
configuration to take effect.
```
- 对于NPU 1.X.X、NPU 20.0.X以及NPU 20.1.X系列版本，如下：
```
npu-smi set -t ssh-enable -i 1 -d 1
Status                  : OK
Message                 : Set ssh-enable successfully, need reset card to active.
```

10.1.4.6.2 配置指定芯片的 ssh 使能状态

命令功能

npu-smi set -t ssh-enable -i *id* **-c** *chip_id* **-d** *value*用于配置指定芯片的ssh使能状态。

命令格式

npu-smi set -t ssh-enable -i *id* **-c** *chip_id* **-d** *value*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。如果有4个芯片，则编号为0、1、2、3。
value	ssh使能模式：分为禁用、使能。 ● 0：禁用 ● 1：使能

使用指南

- 虚拟机和docker容器中不支持该命令。
- 该命令仅支持在root用户下执行。
- 配置指定芯片的ssh使能状态后，需要用户手动复位芯片生效配置，具体请参见**10.1.4.6.7 复位芯片**。
- 打开ssh服务之后，请务必立即修改Device的ssh登录密码（包括HwHiAiUser用户和root用户的密码）。
- 由于配置持久化功能，打开ssh服务之后不会自动关闭，建议在使用完成后及时主动关闭ssh服务。

使用实例

#禁用设备1中编号为0的芯片的ssh使能。

- 对于NPU 20.2.X及以上系列版本，如下：
```
npu-smi set -t ssh-enable -i 1 -c 0 -d 0
Status                  : OK
Message                 : The ssh-enable of the chip is set successfully. Reset chip for the
configuration to take effect.
```
- 对于NPU 1.X.X、NPU 20.0.X以及NPU 20.1.X系列版本，如下：
```
npu-smi set -t ssh-enable -i 1 -c 0 -d 0
Status                  : OK
Message                 : Set ssh-enable successfully, need reset chip to active.
```

10.1.4.6.3 配置 ECC 使能状态

命令功能

npu-smi set -t ecc-enable -i *id* **-d** *value* 用于配置所有芯片的ECC使能状态。

命令格式

npu-smi set -t ecc-enable -i *id* **-d** *value*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
value	ECC模式：分为禁用、使能。 ● 0：禁用 ● 1：使能

使用指南

- 虚拟机和docker容器中不支持该命令。
- 该命令仅支持在root用户下执行。
- 配置所有芯片的ECC使能状态后，需要用户手动复位标卡生效配置，具体请参见**10.1.4.6.6 复位标卡**。
- ECC模式默认为禁用，若配置为使能状态后，会使某些性能下降，请谨慎开启。

使用实例

#开启设备1的ECC使能。

- 对于NPU 20.2.X及以上系列版本，如下：
```
npu-smi set -t ecc-enable -i 1 -d 1
Status                  : OK
Message                 : The ecc-enable is set successfully. Reset card for the configuration to take
effect.
```
- 对于NPU 1.X.X、NPU 20.0.X以及NPU 20.1.X系列版本，如下：
```
npu-smi set -t ecc-enable -i 1 -d 1
Status                  : OK
Message                 : Set ecc-enable successfully
```

10.1.4.6.4 配置 p2p 使能状态

命令功能

npu-smi set -t p2p-enable -i *id* **-d** *value* 用于配置所有芯片的p2p使能状态。

命令格式

npu-smi set -t p2p-enable -i *id* **-d** *value*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
value	p2p模式：分为禁用、使能。 ● 0：禁用 ● 1：使能

使用指南

- 对于NPU 1.X.X系列版本，暂不支持配置芯片的p2p模式为使能，支持配置芯片的p2p模式为禁用。
- 对于NPU 20.X.X及以上系列版本，没有此命令。
- 虚拟机和docker容器中不支持该命令。
- 该命令仅支持在root用户下执行。

使用实例

#禁用设备1的p2p使能。

```
npu-smi set -t p2p-enable -i 1 -d 0
Status                      : OK
Message                     : Set p2p-enable successfully
```

10.1.4.6.5 收集日志

命令功能

npu-smi set -t collect-log -i *id*用于收集MCU的日志。对于NPU 1.X.X系列和NPU 20.0.0版本，收集的日志存放在/tmp目录下；对于NPU 20.0.0以上、NPU 20.1.X和NPU 20.2.X及以上系列版本，收集的日志存放在/run目录下。

命令格式

npu-smi set -t collect-log -i *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

- 虚拟机和docker容器中不支持该命令。
- 该命令仅支持在root用户下执行。

使用实例

#收集设备1的MCU的日志。

- 对于NPU 20.2.X及以上系列版本，如下：
```
npu-smi set -t collect-log -i 1
type(0): file_len(203198)--offset(203198) [100].
Logs collected successfully,(0).
type(1): file_len(101991)--offset(101991) [100].
Logs collected successfully,(1).
type(2): file_len(101376)--offset(101376) [100].
Logs collected successfully,(2).
```
- 对于NPU 1.X.X、NPU 20.0.X以及NPU 20.1.X系列版本，如下：
```
npu-smi set -t collect-log -i 1
type(0): file_len(203198)--offset(203198) [100].
collect log success,(0).
type(1): file_len(101991)--offset(101991) [100].
collect log success,(1).
type(2): file_len(101376)--offset(101376) [100].
collect log success,(2).
```

10.1.4.6.6 复位标卡

软件准备

带外标卡复位功能依赖ipmitool软件，需要提前下载并加载驱动。

1. 下载ipmitool软件：
   - CentOS和EulerOS：**yum install ipmitool**
   - Ubuntu：**apt-get install ipmitool**
2. 加载驱动：

   modprobe ipmi_si

   modprobe ipmi_devintf

   modprobe ipmi_msghandler
3. 复位OS后，需要查看ipmi.service服务状态是否为“enabled”，如果不是，执行2重新加载驱动。

命令功能

npu-smi set -t reset -i *id* [**-m 1**]用于复位PCIe卡。

命令格式

npu-smi set -t reset -i *id* [**-m 1**]

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

- 虚拟机和docker容器中不支持该命令。
- 该命令仅支持在root用户下执行。
- 业务运行过程中，请不要执行此命令。
- 以上命令中[**-m 1**]表示可选参数，如果使用该参数表示通过带内复位标卡，否则表示通过带外复位标卡。该参数仅支持NPU 20.1.X及以上系列版本使用。
- 带外复位标卡需要依赖iBMC，为保障复位功能，请升级服务器iBMC到最新的版本。
- 标卡复位成功后，请至少等待20s，再执行当前或其他标卡、芯片信息查询操作，请耐心等待。

使用实例

#通过带外复位设备1。

- 对于NPU 20.2.X及以上系列版本，如下：
```
npu-smi set -t reset -i 1
Resetting the standard PCIe card or NPU chip may cause a system hang or abnormal reset during service running.
Are you sure you want to continue resetting?(Y/N)
y
Message                    : resetting ...
Status                     : OK
Message                    : Reset card successfully.
```
- 对于NPU 1.X.X和NPU 20.0.X和NPU 20.1.X系列版本，如下：
```
npu-smi set -t reset -i 1
Resetting a standard PCIe card or npu chip during service running may cause system hang or abnormal reset.
Are you sure you want to continue resetting?(Y/N)
y
Message                    : resetting ..
Status                     : OK
Message                    : Reset card successfully
```

#通过带内复位设备77。

- 对于NPU 20.2.X及以上系列版本，如下：
```
npu-smi set -t reset -i 77 -m 1
Resetting the standard PCIe card or NPU chip may cause a system hang or abnormal reset during service running.
Are you sure you want to continue resetting?(Y/N)
y
Message                    : resetting ...
Status                     : OK
Message                    : Reset card successfully.
```
- 对于NPU 20.1.X系列版本，如下：
```
npu-smi set -t reset -i 77 -m 1
Resetting a standard PCIe card or npu chip during service running may cause system hang or abnormal reset.
```

```
Are you sure you want to continue resetting?(Y/N)
y
Message                    : resetting ...
Status                     : OK
Message                    : Reset card successfully
```

10.1.4.6.7 复位芯片

软件准备

带外芯片复位功能依赖ipmitool软件，需要提前下载并加载驱动。

1. 下载ipmitool软件：
   - CentOS和EulerOS：**yum install ipmitool**
   - Ubuntu：**apt-get install ipmitool**
2. 加载驱动：

   modprobe ipmi_si

   modprobe ipmi_devintf

   modprobe ipmi_msghandler
3. 复位OS后，需要查看ipmi.service服务状态是否为“enabled”，如果不是，执行2重新加载驱动。

命令功能

npu-smi set -t reset -i *id* **-c** *chip_id* [**-m 1**]用于复位指定芯片。

命令格式

npu-smi set -t reset -i *id* **-c** *chip_id* [**-m 1**]

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。如果有4个芯片，则编号为0、1、2、3。

使用指南

- 虚拟机和docker容器中不支持该命令。
- 该命令仅支持在root用户下执行。
- 业务运行过程中，请不要执行此命令。
- 以上命令中[**-m 1**]表示可选参数，如果使用该参数表示通过带内复位指定芯片，否则表示通过带外复位指定芯片。该参数仅NPU 20.1.X及以上系列版本使用。
- 带外复位芯片需要依赖iBMC，为保障复位功能，请升级服务器iBMC到最新的版本。
- 芯片复位成功后，请至少等待20s，再执行当前或其他芯片信息查询操作，请耐心等待。

- 该命令不支持指定MCU带外复位。

使用实例

#通过带外复位设备1编号为0的芯片。

- 对于NPU 20.2.X及以上系列版本，如下：

```
npu-smi set -t reset -i 1 -c 0
Resetting the standard PCIe card or NPU chip may cause a system hang or abnormal reset during
service running.
Are you sure you want to continue resetting?(Y/N)
y
Message                    : resetting ...
Status                     : OK
Message                    : Reset chip successfully.
```

- 对于NPU 1.X.X、NPU 20.0.X和NPU 20.1.X系列版本，如下：

```
npu-smi set -t reset -i 1 -c 0
Resetting a standard PCIe card or npu chip during service running may cause system hang or
abnormal reset.
Are you sure you want to continue resetting?(Y/N)
y
Message                    : resetting ...
Status                     : OK
Message                    : Reset chip successfully
```

#通过带内复位设备77编号为0的芯片。

- 对于NPU 20.2.X及以上系列版本，如下：

```
npu-smi set -t reset -i 77 -c 0 -m 1
Resetting the standard PCIe card or NPU chip may cause a system hang or abnormal reset during
service running.
Are you sure you want to continue resetting?(Y/N)
y
Message                    : resetting ...
Status                     : OK
Message                    : Reset chip successfully.
```

- 对于NPU 20.1.X系列版本，如下：

```
npu-smi set -t reset -i 77 -c 0 -m 1
Resetting a standard PCIe card or npu chip during service running may cause system hang or
abnormal reset.
Are you sure you want to continue resetting?(Y/N)
y
Message                    : resetting ...
Status                     : OK
Message                    : Reset chip successfully
```

10.1.4.6.8 配置用户证书

命令功能

npu-smi set -t license -i *id* **-f** "*license*"用于配置用户的证书。证书只能配置一次，否则会配置失败。

命令格式

npu-smi set -t license -i *id* **-f** "*license*"

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
license	证书内容。 格式为字符串，大小为1-255个字节。 **说明** 由于Linux命令行的Shell对部分特殊字符具有保留含义，如**$$**表示Shell进程ID，**$?**表示上个命令的退出状态，故如需要设置特殊字符，用户需要对特殊字符进行转义操作。

使用指南

- 对于NPU 1.X.X系列版本：该命令仅支持npu-smi及MCU版本为1.4.0及以上版本。
- 对于NPU 20.X.X及以上系列版本：所有版本均支持此命令。
- 虚拟机和docker容器中不支持该命令。
- 该命令仅支持在root用户下执行。

使用实例

#配置设备1的证书为abcde。

- 对于NPU 20.2.X及以上系列版本，如下:
```
npu-smi set -t license -i 1 -f "abcde"
Status                  : OK
Message                   : The license of the chip is set successfully.
```
- 对于NPU 1.X.X、NPU 20.0.X以及NPU 20.1.X系列版本，如下：
```
npu-smi set -t license -i 1 -f "abcde"
Status                  : OK
Message                   : Set chip license successfully
```

10.1.4.6.9 设置 AI 芯片的 IP 信息

命令功能

npu-smi set -t ip -i *id* **-c** *chip_id* **-s** *ipstring* 用于设置AI芯片的IP信息。

命令格式

npu-smi set -t ip -i *id* **-c** *chip_id* **-s** *ipstring*

参数说明

类型	描述
id	设备id。 通过**npu-smi info -l**命令查出的NPU ID即为设备id。

类型	描述
chip_id	芯片id。 如果有4个芯片，则编号为0、1、2、3。
ipstring	IP地址及子网掩码。 格式如下：“XX.XX.XX.XX/XX.XX.XX.XX” 其中，“/”前半部分为IP地址，后半部分为子网掩码。

使用指南

- 对于NPU 1.X.X系列版本：仅NPU 1.4.1及以上版本支持此命令。
- 对于NPU 20.X.X系列版本，所有版本均支持此命令。
- 虚拟机和docker容器中不支持该命令。
- 该命令仅支持在root用户下执行。
- 重启系统和复位芯片后配置的IP信息都会丢失，恢复为默认IP地址。需要重新设置AI芯片的IP信息。
- 配置IP地址必须在物理机的root用户下运行，若在物理机的非root用户，或在容器、虚拟机下运行，则会返回错误。
- 对于NPU 1.X.X系列版本，配置IP地址必须先使能芯片的ssh功能，否则只会修改芯片的IP信息，ssh服务重启失败，从而导致返回错误。

使用实例

#设置设备0，编号为0的AI芯片的IP信息为192.168.5.199，子网掩码为255.255.255.0。

- 对于NPU 20.2.X及以上系列版本，如下：
```
npu-smi set -t ip -i 0 -c 0 -s 192.168.5.199/255.255.255.0
Status                  : OK
Message                   : The ip of the chip is set successfully.
```
- 对于NPU 1.X.X、NPU 20.0.X以及NPU 20.1.X系列版本，如下：
```
npu-smi set -t ip -i 0 -c 0 -s 192.168.5.199/255.255.255.0
Status                  : OK
Message                   : Set chip ip successfully
```

10.1.4.6.10 清除所有芯片的 PCIE 误码计数

命令功能

npu-smi set -t errcount-clear -i *id* 用于清除所有芯片PCIE误码计数。

命令格式

npu-smi set -t errcount-clear -i *id*

参数说明

类型	描述
id	设备id。 通过**npu-smi info -l**命令查出的NPU ID即为设备id。

使用指南

- 该命令仅支持NPU 20.1.X及以上系列版本。
- 虚拟机和docker容器中不支持该命令。
- 该命令仅支持在root用户下执行。

使用实例

#清除设备1所有芯片的PCIE误码计数。

- 对于NPU 20.2.X及以上系列版本，如下：
```
npu-smi set -t errcount-clear -i 1
Status                      : OK
Message                       : The errcount-clear is set successfully.
```
- 对于NPU 20.1.X系列版本，如下：
```
npu-smi set -t errcount-clear -i 1
Status                      : OK
Message                       : Set errcount-clear successfully
```

10.1.4.6.11 清除指定芯片的 PCIE 误码计数

命令功能

npu-smi set -t errcount-clear -i *id* **-c** *chip_id* 用于清除指定芯片PCIE误码计数。

命令格式

npu-smi set -t errcount-clear -i *id* **-c** *chip_id*

参数说明

类型	描述
id	设备id。 通过**npu-smi info -l**命令查出的NPU ID即为设备id。
chip_id	芯片id。如果有4个芯片，则编号为0、1、2、3。

使用指南

- 该命令仅支持NPU 20.1.X及以上系列版本。
- 虚拟机和docker容器中不支持该命令。

- 该命令仅支持在root用户下执行。

使用实例

#清除设备1中编号为1的芯片的PCIE误码计数。

- 对于NPU 20.2.X及以上系列版本，如下：
```
npu-smi set -t errcount-clear -i 1 -c 1
Status                  : OK
Message                 : The errcount-clear of the chip is set successfully.
```
- 对于NPU 20.1.X系列版本，如下：
```
npu-smi set -t errcount-clear -i 1 -c 1
Status                  : OK
Message                 : Set chip errcount-clear successfully
```

10.1.4.7 升级功能（upgrade）

10.1.4.7.1 查询固件升级状态

命令功能

npu-smi upgrade -q *item* **-i** *id* 用于查询固件的升级状态。

命令格式

npu-smi upgrade -q *item* **-i** *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
item	升级类型：MCU。

使用指南

- 虚拟机和docker容器中不支持该命令。
- 该命令仅支持在root用户下执行。

使用实例

#查询设备1中MCU的固件升级状态。

- 对于NPU 20.2.X及以上系列版本，如下：
```
npu-smi upgrade -q mcu -i 1
Conclusion              : PASS
Message                 : The device is upgraded successfully.
```
- 对于NPU 1.X.X、NPU 20.0.X以及NPU 20.1.X系列版本，如下：
```
npu-smi upgrade -q mcu -i 1
Conclusion              : PASS
Message                 : Device upgrade successfully.
```

10.1.4.7.2 查询固件版本信息

命令功能

npu-smi upgrade -b *item* **-i** *id* 用于查询固件的版本信息。

命令格式

npu-smi upgrade -b *item* **-i** *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
item	升级类型：MCU。

使用指南

虚拟机和docker容器中不支持该命令。

使用实例

#查询设备1中MCU的固件版本信息。

```
npu-smi upgrade -b mcu -i 1
Version                       : 1.1.8
```

10.1.4.7.3 升级固件

命令功能

npu-smi upgrade -t *item* **-i** *id* **-f** *file_path*用于启动固件升级。

命令格式

npu-smi upgrade -t *item* **-i** *id* **-f** *file_path*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
item	升级类型：MCU。
file_path	升级文件路径。 文件路径仅支持英文字母、数字、下划线("_")、点(".")、斜线("/")、连字符("-")。

使用指南

- 虚拟机和docker容器中不支持该命令。
- 该命令仅支持在root用户下执行。

使用实例

#升级设备1中的MCU固件。

- 对于NPU 20.2.X及以上系列版本，如下：

```
npu-smi upgrade -t mcu -i 1 -f ./IT91VPUA_MCU_V103.bin
        Validity                    : success

file_len(160847)--offset(160847) [100].
        Transfer                    : success
        Status                      : start to upgrade
Start upgrade [100].
        Status                      : OK
        Message                     : The device upgrade is started successfully
        Message                     : need active mcu
```

- 对于NPU 1.X.X、NPU 20.0.X以及NPU 20.1.X系列版本，如下：

```
npu-smi upgrade -t mcu -i 1 -f ./IT91VPUA_MCU_V103.bin
        validate                    : successfully

file_len(160847)--offset(160847) [100].
        transfile                   : successfully
        Status                      : start to upgrade
Start upgrade [100].
        Status                      : OK
        Message                     : Start device upgrade successfully
        Message                     : need active mcu
```

10.1.4.7.4 生效固件

命令功能

npu-smi upgrade -a *item* **-i** *id* 在固件升级完成之后，用该命令生效固件的新版本。

命令格式

npu-smi upgrade -a *item* **-i** *id*

参数说明

类型	描述
id	设备id。通过**npu-smi info -l**命令查出的NPU ID即为设备id。
item	升级类型：MCU。

使用指南

- 虚拟机和docker容器中不支持该命令。
- 该命令仅支持在root用户下执行。

使用实例

#生效设备1中MCU的新版本固件。

- 对于NPU 20.2.X及以上系列版本，如下：
```
npu-smi upgrade -a mcu -i 1
Status                  : OK
Message                 : The upgrade has taken effect after performed reboot successfully.
```
- 对于NPU 1.X.X、NPU 20.0.X以及NPU 20.1.X系列版本，如下：
```
npu-smi upgrade -a mcu -i 1
Status                  : OK
Message                 : Active upgrade warm reboot successfully.
```

10.1.5 卸载 npu-smi 工具

npu-smi工具的rpm包格式如下：A800-3000-NPU-EulerOS2.8-Dsmitool-ARM-Vx.x.x.rpm。本章以A800-3000-NPU-EulerOS2.8-Dsmitool-ARM-V1.4.1.rpm为例介绍npu-smi工具卸载的相关指导。run包的驱动卸载后会自动卸载npu-smi工具。

前提条件

已安装npu-smi工具。具体请执行**npu-smi -v**命令，若能查询到工具版本信息，则表示已安装。

操作步骤

步骤1 使用PuTTy登录Atlas 300I 推理卡（型号 3000）所在服务器的OS命令行，具体请参见**9.9.5 通过PuTTY登录（网口方式）**。

步骤2 执行如下命令，切换至root用户。

su - root

步骤3 执行如下命令查看npu-smi的rpm包列表。

 # rpm -qa | grep npu

回显如下，表示存在npu-smi工具。

```
npu-smi-1.4.1-1.aarch64
```

步骤4 执行如下命令卸载npu-smi工具。

 # rpm -e npu-smi-1.4.1-1.aarch64

说明

以上命令中的**npu-smi-1.4.1-1.aarch64**为**步骤3**查询到的工具包名，请根据实际情况替换。

步骤5 卸载完成后，执行**npu-smi** *command*命令若无法使用，表示卸载成功。

步骤6 如果需要再次使用npu-smi工具，请重新安装。具体请参见**10.1.1 安装npu-smi工具**。

----结束

10.2 NPU 一键式日志收集工具

须知

- 本章节内容仅适用于NPU 21.0.3版本。
- NPU一键式日志收集工具不支持容器场景。
- 建议用户在root权限下使用该工具收集日志信息，避免收集不到部分日志。

操作步骤

步骤1 获取软件包。详细请参考**9.3 获取软件包**。

步骤2 安装NPU一键式日志收集工具。

1. 登录Host侧服务器。详细请参考**9.9.5 通过PuTTY登录（网口方式）**。
2. 安装驱动。详细请参考**9.5 安装驱动**。
3. 获取NPU一键式日志收集工具。
   驱动包在安装过程中会默认安装NPU一键式日志收集工具（log_collect.sh）。

   说明

   使用NPU 21.0.3软件包安装完成后，log_collect.sh默认安装路径为“/usr/local/bin/”和“/usr/local/sbin/”。

步骤3 执行如下命令，收集日志信息。

log_collect.sh

NPU一键式日志收集工具主要收集如下信息：

存储目录	说明
collect_all_log/device_log/	Device日志。
collect_all_log/npu_log/	NPU设备日志。
collect_all_log/os_log/	OS日志。
collect_all_log/mcu_log/	MCU日志。
collect_all_log/collect_result.log	记录收集日志结果。
collect_all_log/var_log/	/var/log目录下部分文件。

说明

- 日志包获取路径如下：
  - root用户：/root/collect_all_log.tar.gz
  - 非root用户：/home/*用户名*/collect_all_log.tar.gz
- 在虚拟机场景下不支持收集部分日志，具体日志信息请查看生成的压缩包“collect_all_log.tar.gz”中的collect_result.log文件。

----结束

10.3 驱动一键式收集日志工具

驱动一键式收集日志工具部署在Host侧，能够一键收集日志信息。

须知

- 本章节内容仅适用于NPU 22.0.0及以上版本。
- 该工具不支持容器场景，不支持多个用户同时运行。
- 该工具需用户在root权限下运行。
- 系统不会自动删除日志文件，为避免日志文件占用磁盘空间，建议用户在下载后手动删除。

操作步骤

步骤1 登录Host侧服务器。

步骤2 获取驱动一键式收集日志工具。

该工具在驱动Driver的安装目录下，路径为“*Driver安装目录*/driver/tools/npu_log_collect.sh”。

步骤3 在具有读、写、执行权限的目录下执行如下命令，运行驱动一键式收集日志工具。

命令： *Driver安装目录***/driver/tools/npu_log_collect.sh** [options]

示例：/usr/local/Ascend/driver/tools/npu_log_collect.sh

其中[options]支持的参数及解释请参见**表10-151**。

表 10-151 参数说明

参数	说明	举例
不指定任何参数	在执行目录下，导出默认收集内容。	*Driver安装目录***/driver/tools/npu_log_collect.sh**
-h或--help	显示帮助信息。	*Driver安装目录***/driver/tools/npu_log_collect.sh -h**

参数	说明	举例
-p或--path	指定生成文件目录，该目录需要具有读、写、执行的权限。	*Driver安装目录*/**driver/tools/npu_log_collect.sh -p** /home/tmp/
--off	关闭收集指定内容。 host_info：Host侧环境信息。 host_log：Host侧日志信息。 device_log：Device侧全量日志。 driver_info：驱动运行信息。 install_info：驱动安装信息。	*Driver安装目录*/**driver/tools/npu_log_collect.sh --off** host_info

运行成功后，在当前路径下会生成一个压缩文件“npu_log_collect_*YYYYMMDDhhmmss*.tar.gz”存储日志信息，文件后缀名为日志收集完成时间戳，解压之后生成表10-152所示存储目录及文件。

表 10-152 日志存储目录文件

日志存储目录及文件	说明
collect_scripts_running_log	日志收集脚本运行日志。
mcu_log	MCU日志。
npu_info_log	NPU状态信息。
nputools_log	npu-smi工具运行日志。
ascend_log_*YYYYMMDDhhmmss*.tar.gz	昇腾芯片运行日志。文件后缀名为调用工具时间戳，解压后目录及文件请参见表10-153。

表 10-153 昇腾芯片运行日志解压后目录文件

日志目录及文件	说明
host_info	当前Host侧环境信息。
host_log	当前Host侧日志信息。
device_log	Device侧全量日志。
driver_info	驱动运行信息。
install_info	驱动安装信息。

日志目录及文件	说明
script_running.log	工具运行日志。

----结束

11 维护管理

Atlas 300I 推理卡（型号 3000）提供了丰富的维护管理功能，包括运行在OS中的带内管理命令集和通过BMC提供的带外管理功能。

说明

如果AI芯片没有加载驱动，则带外管理无法准确识别AI芯片是否真正发生故障，因此带外对AI芯片失效场景不做告警提示。带内管理只提供AI芯片健康状况的查询，如果上层业务需要对AI芯片失效场景做实时告警，需要上层业务调用DCMI API中相关接口，并做相关处理。

11.1 带内管理

11.2 带外管理

11.1 带内管理

带内管理的功能有：

- 在线升级功能，升级Firmware，方便用户的设备维护。
- 资产管理功能，提供序列号等信息，方便用户进行资产管理。
  具体资产管理操作请参见**10.1 npu-smi工具**。

11.2 带外管理

Atlas 300I 推理卡（型号 3000）提供SMBUS接口，支持服务器的带外管理功能。BMC提供带外管理功能，包括资产信息及监控Atlas 300I 推理卡（型号 3000）温度、电压、实时功耗及芯片监控状态等信息。同时BMC能够接管Atlas 300I 推理卡（型号 3000）的对应告警信息。

具体带外管理接口请参见**《Atlas 300I 推理卡 带外管理接口说明（型号 3000, 3010）》**。

12 容器内安装

12.1 简介
12.2 宿主机目录挂载容器
12.3 宿主机目录不挂载容器
12.4 容器内软件包卸载

12.1 简介

本章内容仅适用于NPU 20.X.X系列及以上版本，介绍软件包在容器内的安装以及卸载的详细流程。使用容器安装软件包之前，请务必查看如下限制：

- 使用NPU 20.0.X和NPU 20.1.X系列版本软件包安装时，docker版本要求大于等于18.03。
- Firmware软件包不支持docker内安装。
- 容器内安装的Driver软件包版本需要与宿主机安装的Driver软件包版本保持一致，不支持独立升级。
- 运行环境中的一个device只能被一个容器使用，只有当使用该device的容器退出后，该device才可以被其他容器使用。
- Driver支持docker内安装，只安装驱动的.so，即安装驱动包的用户态部分。根据宿主机目录（默认路径为：/usr/local/Ascend/driver/lib64/ ）是否挂载到容器，分为如下两种安装方式，两种方式二选一，用户根据实际情况进行选择。
  - 将宿主机的目录挂载到容器中，请参见**12.2 宿主机目录挂载容器**。
  - 宿主机的目录不挂载容器，请参见**12.3 宿主机目录不挂载容器**。
- 使用NPU 20.0.X和NPU 20.1.X系列版本软件包安装时，启动docker中的用户必须与host侧的运行用户一致。
- 使用NPU 21.0.X系统及以下版本软件包安装时，不支持源码驱动包在docker容器内安装。

宿主机与容器操作系统兼容性关系

宿主机操作系统与容器操作系统的兼容性关系如**表12-1**。

表 12-1 兼容性关系

操作系统架构	宿主机操作系统版本	宿主机操作系统内核版本	容器操作系统版本	说明
ARM	Ubuntu 18.04.1	4.15.0-29-generic	Ubuntu 18.04	容器OS目前支持的Ubuntu操作系统版本为18.04.1、18.04.4和18.04.5。
			CentOS 7.6	NA
			CentOS 7.8	不支持使用宿主机目录挂载容器方式安装Driver软件包。
	CentOS 7.6	4.14.0-115.el7a.0.1.aarch64	CentOS 7.6	NA
			CentOS 7.8	NA
			Ubuntu 18.04	容器OS目前支持的Ubuntu操作系统版本为18.04.1、18.04.4和18.04.5。
	统信服务器操作系统V20（1020e）	4.19.90-2106.3.0.0095.up2.uel20.aarch64	Ubuntu 18.04.5	NA
			CentOS 7.6	NA
	Ubuntu 20.04	5.4.0-26-generic	Ubuntu 18.04.5	NA
			CentOS 7.6	NA
	EulerOS 2.10	4.19.90-vhulk2111.1.0.h963.eulerosv2r10.aarch64	EulerOS 2.10	NA

说明

表12-1中的容器操作系统为已验证过的版本，具体请以实际获取的操作系统镜像版本为准。

12.2 宿主机目录挂载容器

前提条件

1. 宿主机已经安装过Driver软件包。详情请参见9.5 安装驱动。
2. 需用户在宿主机自行安装docker（使用NPU 20.0.X和NPU 20.1.X系列版本软件包安装时，版本要求大于等于18.03）。

版权所有 © 华为技术有限公司

3. 重启docker服务，命令为：
```
systemctl daemon-reload
systemctl restart docker
```
4. 获取OS镜像，例如：容器镜像为Ubuntu 18.04，命令为：
```
docker pull ubuntu:18.04
```
如果获取的OS镜像版本不正确，建议从**Ascend Hub**上获取正确的OS镜像版本。
5. 查看镜像，命令为：
```
docker images
```

安装步骤

1. 在宿主机使用**id HwHiAiUser**命令查看宿主机HwHiAiUser的gid，并记录该gid的取值，如图12-1所示，则记录gid为1001。

   使用NPU 20.1.X及以上系列版本软件包安装时，如果用户安装Driver包时使用**--install-username=*username* --install-usergroup=*usergroup***参数指定了其他非root用户，则需要使用同样的方法记录此非root用户的gid，容器内创建非root用户时，需要使用该gid，确保宿主机和容器内相关进程的非root用户同属组。

   图 12-1 查看宿主机 HwHiAiUser 的 gid

   ```
   root@ascend-HP-ProDesk-600-G4-PCI-MT:~# id HwHiAiUser
   uid=1027(HwHiAiUser) gid=1001(HwHiAiUser) groups=1001(HwHiAiUser)
   ```

2. 在宿主机创建并启动docker容器。

   需要将表12-2所示参数挂载到容器：

   表 12-2 参数解释

参数	参数说明
--device	表示映射的设备。
-v /usr/local/dcmi:/usr/local/dcmi	将宿主机dcmi的.so和接口文件目录“/usr/local/dcmi”挂载到容器中，请根据实际情况修改。
-v /usr/local/sbin/npu-smi:/usr/local/sbin/npu-smi	使用NPU 20.0.0版本软件包安装时，将宿主机npu-smi工具“/usr/local/sbin/npu-smi”挂载到容器中，请根据实际情况修改。
-v /usr/local/bin/npu-smi:/usr/local/bin/npu-smi	使用NPU 20.0.0以上、NPU 20.1.X及以上系列版本软件包安装时，将宿主机npu-smi工具“/usr/local/bin/npu-smi”挂载到容器中，请根据实际情况修改。
-v /lib64:/lib64	将宿主机目录“/lib64”挂载到容器。 **说明** 驱动版本号为1.73.5.2.B050及以上版本不需要在容器挂载此目录。
-v /var/log/npu/slog/container/container_name:/var/log/npu/slog	使用NPU 20.0.0及以上、NPU 20.1.X系列版本软件包安装时，将宿主机日志路径“/var/log/npu/slog/container/container_name”挂载到容器中。请根据实际情况修改。

版权所有 © 华为技术有限公司

参数	参数说明
-v /var/log/npu/conf/slog/slog.conf:/var/log/npu/conf/slog/slog.conf	使用NPU 20.0.0及以上、NPU 20.1.X系列版本软件包安装时，将宿主机日志配置文件“/var/log/npu/conf/slog/slog.conf”挂载到容器中。请根据实际情况修改。
-v /usr/local/Ascend/driver/lib64/ :/usr/local/Ascend/driver/lib64/	将宿主机目录“/usr/local/Ascend/driver/lib64/ ”挂载到容器，请根据drive的驱动.so所在路径修改。
-v /usr/local/Ascend/driver/tools/:/usr/local/Ascend/driver/tools/	使用NPU 20.0.0及以上、NPU 20.1.X系列版本软件包安装时，将宿主机目录“/usr/local/Ascend/driver/tools/”挂载到容器，请根据slogd所在路径修改。
-v /etc/ascend_install.info:/etc/ascend_install.info	将宿主机安装信息文件“/etc/ascend_install.info”挂载到容器中。
-v /usr/local/Ascend/driver/version.info:/usr/local/Ascend/driver/version.info	将宿主机版本信息文件“/usr/local/Ascend/driver/version.info”挂载到容器中，请根据实际情况修改。

容器启动后，执行以下命令查看当前docker容器中可以使用的davinci设备：

```
ls /dev/ | grep davinci*
```

命令执行情况请参见图12-2。

图 12-2 查询 docker 容器中使用的 davinci 设备

```
root@ascend-HP-ProDesk-600-G4-PCI-MT:/# ls /dev/ | grep davinci*
davinci0
davinci_manager
```

其中：

- davinci_manager为管理模块的字符设备节点。
- davinci0为该容器使用的davinci设备。

3. 在容器内创建运行用户（使用NPU 20.1.X及以上系列版本软件包安装时，运行用户为非root用户，例如：HwHiAiUser），用于启动相关进程。**若运行用户为root用户，请跳过此步骤。**

   ```
   groupadd -g gid HwHiAiUser && useradd -g HwHiAiUser -d /home/HwHiAiUser -m HwHiAiUser && echo ok
   ```

   请将上述命令中的gid替换为1中查出的gid值。若返回“ok”，则说明创建成功。

   如果用户创建了其他非root用户，请确保该用户所属的属组必须和Driver运行用户所属属组相同；如果不同，请用户自行添加到Driver运行用户属组。

4. 使用NPU 20.2.X及以上系列版本软件包安装时，仅需执行步骤**•使用NPU 20.2.X系列版本软件包安装时，...**设置环境变量。使用NPU 20.0.X和NPU 20.1.X系统版本软件包安装时，需要执行以下步骤启动slogd守护进程。

a. 在容器内设置环境变量（通过export方式设置环境变量，该种方式设置的环境变量只在当前窗口有效，设置完成后立即生效）：

- 使用NPU 20.0.X和NPU 20.1.X系统版本软件包安装时，执行如下命令：
```
export LD_LIBRARY_PATH=/usr/local/Ascend/driver/lib64:/usr/local/Ascend/add-ons:$LD_LIBRARY_PATH
```
- 使用NPU 20.2.X及以上系列版本软件包安装时，执行如下命令：
```
export LD_LIBRARY_PATH=/usr/local/Ascend/driver/lib64:$LD_LIBRARY_PATH
```
其中“/usr/local/Ascend”为默认安装路径，请根据实际情况修改。

b. 执行如下命令创建日志相关文件，并启动slogd进程。
```
mkdir -p /usr/slog
mkdir -p /var/log/npu/slog/slogd
chown -Rf HwHiAiUser:HwHiAiUser  /usr/slog
chown -Rf HwHiAiUser:HwHiAiUser  /var/log/npu/slog
su HwHiAiUser --command "/usr/local/Ascend/driver/tools/slogd &"
```

c. 执行如下命令查看slogd进程，确认驱动包是否安装成功。
```
ps -ef | grep -v grep | grep "tools/slogd"
```
若返回如下信息，则说明进程已启动，驱动包安装成功。
```
root@ascend-HP-ProDesk-600-G4-PCI-MT:/var/log/npu/conf/slog# ps -ef | grep -v grep | grep "tools/slogd"
HwHiAiU+    38     1  0 07:03 ?        00:00:00 /usr/local/Ascend/driver/tools/slogd
```

12.3 宿主机目录不挂载容器

前提条件

1. 宿主机已经安装过Driver软件包。详情请参见**9.5 安装驱动**。
2. 需用户在宿主机自行安装docker（使用NPU 20.0.X和NPU 20.1.X系列版本软件包安装时，版本要求大于等于18.03）。
3. 重启docker服务，命令为：
```
systemctl daemon-reload
systemctl restart docker
```
4. 获取OS镜像，例如：容器镜像为Ubuntu 18.04，命令为：
```
docker pull ubuntu:18.04
```
如果获取的OS镜像版本不正确，建议从**Ascend Hub**上获取正确的OS镜像版本。
5. 查看镜像，命令为：
```
docker images
```

操作步骤

1. 在宿主机使用**id HwHiAiUser**命令查看宿主机HwHiAiUser的gid，并记录该gid的取值，如**图12-3**所示，则记录gid为1001。

   使用NPU 20.1.X及以上系列版本软件包安装时，如果用户安装Driver包时使用**--install-username=*username* --install-usergroup=*usergroup***参数指定了其他非root用户，则需要使用同样的方法记录此非root用户的gid，容器内创建非root用户时，需要使用该gid，确保宿主机和容器内相关进程的非root用户同属组。

   图 12-3 查看宿主机 HwHiAiUser 的 gid

```
root@ascend-HP-ProDesk-600-G4-PCI-MT:~# id HwHiAiUser
uid=1027(HwHiAiUser) gid=1001(HwHiAiUser) groups=1001(HwHiAiUser)
```

2. 在宿主机创建并启动docker容器。

   需要将表12-3所示参数挂载到容器：

   表 12-3 参数解释

参数	参数说明
--device	表示映射的设备。
-v /usr/local/dcmi:/usr/local/dcmi	将宿主机dcmi的.so和接口文件目录“/usr/local/dcmi”挂载到容器中，请根据实际情况修改。
-v /usr/local/sbin/npu-smi:/usr/local/sbin/npu-smi	使用NPU 20.0.0版本软件包安装时，将宿主机npu-smi工具“/usr/local/sbin/npu-smi”挂载到容器中，请根据实际情况修改。
-v /usr/local/bin/npu-smi:/usr/local/bin/npu-smi	使用NPU 20.0.0以上、NPU 20.1.X及以上系列版本软件包安装时，将宿主机npu-smi工具“/usr/local/bin/npu-smi”挂载到容器中，请根据实际情况修改。
-v /var/log/npu/slog/container/container_name:/var/log/npu/slog	使用NPU 20.0.0及以上、NPU 20.1.X及以上系列版本软件包安装时，将宿主机日志路径“/var/log/npu/slog/container/container_name”挂载到容器中。请根据实际情况修改。
-v /var/log/npu/conf/slog/slog.conf:/var/log/npu/conf/slog/slog.conf	使用NPU 20.0.0及以上、NPU 20.1.X及以上系列版本软件包安装时，将宿主机日志配置文件“/var/log/npu/conf/slog/slog.conf”挂载到容器中。请根据实际情况修改。

3. 在容器内创建运行用户（使用NPU 20.1.X及以上系列版本软件包安装时，运行用户为非root用户，例如：HwHiAiUser），用于启动相关进程。**若运行用户为root用户，请跳过此步骤。**

   ```
   groupadd -g gid HwHiAiUser && useradd -g HwHiAiUser -d /home/HwHiAiUser -m HwHiAiUser &&
   echo ok
   ```

   请将上述命令中的gid替换为1中查出的gid值。若返回“ok”，则说明创建成功。

   如果用户创建了其他非root用户，请确保该用户所属的属组必须和Driver运行用户所属属组相同；如果不同，请用户自行添加到Driver运行用户属组。

4. 使用**exit**命令退出容器，在宿主机Driver软件包所在路径，执行如下命令将软件包复制到容器内部。

   ```
   docker cp /xxx/*.run container_id:/home/HwHiAiUser/software
   ```

   其中xxx为宿主机上软件包的存放路径，*.run请替换为具体软件包名，*container_id*为容器ID，可以使用**docker ps -a**命令查看所使用容器的ID，“*/home/HwHiAiUser/software*”为容器内软件包的存放路径，如果没有该路径，请先手动创建；所有路径请用户根据实际情况进行修改。

5. 使用如下命令重新进入容器，并进入*software*目录，根据表12-4所示安装命令以及安装用户安装驱动软件包。

   ```
   docker start xxx
   docker attach xxx
   ```

   其中xxx请替换为具体容器ID或容器名。

表 12-4 安装命令

组件	安装命令	安装用户	运行用户	是否支持安装多版本	备注
Driver	./*.run --docker 默认路径：/usr/local/Ascend 如果用户需要指定安装路径，需要通过**--install-path=<path>**参数指定。	root	HwHiAiUser 或通过**--install-username=*username* --install-usergroup=*usergroup***参数指定非root用户	否	更多安装模式请参见**9.11.1 参数说明**。

若安装过程中无错误信息提示，则表示安装成功。

- 软件包默认安装路径：/usr/local/Ascend
- 安装详细日志路径：/var/log/ascend_seclog/ascend_install.log
- 安装后软件包的安装路径、安装命令以及运行用户信息记录路径：/etc/ascend_install.info

6. 使用NPU 20.2.X及以上系列版本软件包安装时，仅需执行步骤**•使用NPU 20.2.X系列版本软件包安装时，...**设置环境变量。使用NPU 20.0.X和NPU 20.1.X系统版本软件包安装时，需要执行以下步骤启动slogd守护进程。

   a. 在容器内设置环境变量（通过export方式设置环境变量，该种方式设置的环境变量只在当前窗口有效，设置完成后立即生效）：

      - 使用NPU 20.0.X和NPU 20.1.X系统版本软件包安装时，执行如下命令：
        ```
        export LD_LIBRARY_PATH=/usr/local/Ascend/driver/lib64:/usr/local/Ascend/add-ons:$LD_LIBRARY_PATH
        ```
      - 使用NPU 20.2.X及以上系列版本软件包安装时，执行如下命令：
        ```
        export LD_LIBRARY_PATH=/usr/local/Ascend/driver/lib64:$LD_LIBRARY_PATH
        ```
        其中“/usr/local/Ascend”为默认安装路径，请根据实际情况修改。

   b. 执行如下命令创建日志相关文件，并启动slogd进程。
      ```
      mkdir -p /usr/slog
      mkdir -p /var/log/npu/slog/slogd
      chown -Rf HwHiAiUser:HwHiAiUser  /usr/slog
      chown -Rf HwHiAiUser:HwHiAiUser  /var/log/npu/slog
      su HwHiAiUser --command "/usr/local/Ascend/driver/tools/slogd &"
      ```

   c. 执行如下命令查看slogd进程，确认驱动包是否安装成功。
      ```
      ps -ef | grep -v grep | grep "tools/slogd"
      ```


版权所有 © 华为技术有限公司

若返回如下信息，则说明进程已启动，驱动包安装成功。

```
root@ascend-HP-ProDesk-600-G4-PCI-MT:/var/log/npu/conf/slog# ps -ef | grep -v grep | grep "tools/slogd"
HwHiAiU+    38     1  0 07:03 ?        00:00:00 /usr/local/Ascend/driver/tools/slogd
```

12.4 容器内软件包卸载

容器内驱动包的卸载与宿主机一样，具体请参见**9.7 卸载驱动**。不支持固件卸载。

版权所有 © 华为技术有限公司

A 附录

A.1 术语

A

AI 研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。

B

BIOS 存于计算机主板上的一种固件。包括基本输入输出控制程序、上电自检程序、系统启动自举程序、系统设置信息，为计算机提供底层的硬件设置和控制功能。

C

CPU 中央处理器是计算机的主要设备之一，其功能是解释计算机指令以及处理计算机软件中的数据，与内部存储器、输入及输出设备成为现代电脑的三大部件。

D

DDK DDK是Mind解决方案提供的开发者套件包，Mind Studio通过安装DDK后获得Mind开发必需的API、库、工具链等开发组件。

DKMS 动态内核模块支持是用来生成Linux内核模块的一个框架，其源代码一般不在Linux内核源代码树。当新的内核安装时，DKMS支持的内核设备驱动程序会自动重建。

H

HDC 用于Host和Device之间通信模块，在Host和Device里面均有部署。

I

IDE 集成开发环境。

N

NPU 采用“数据驱动并行计算”的架构，特别擅长处理视频、图像类的海量多媒体业数据，专门用于处理人工智能应用中的大量计算任务。

P

PCIe	PCIe属于高速串行点对点双通道高带宽传输，所连接的设备分配独享通道带宽，不共享总线带宽，主要支持主动电源管理，错误报告，端对端的可靠性传输，热插拔以及服务质量(QOS)等功能。
R	
Runtime	Runtime运行于APP进程空间，为APP提供了Ascend 310设备的Memory管理、Device管理、Stream管理、Event管理、Kernel执行等功能。
T	
TE	用于开发自定义算子。
TEE	在Arm Trustzone的硬件隔离环境基础上，结合硬件可信根设计，实现安全启动、安全存储、安全升级、安全运行等功能，为系统提供可信的基础运行环境。

A.2 缩略语

A		
AI	Artificial Intelligence	人工智能
B		
BMC	Baseboard Management Controller	主板管理控制单元
C		
CFM	Cubic Feet Per Minute	立方英尺每分钟
D		
DCMI	DaVinci Card Management Interface	达芬奇卡管理接口
M		
MCU	Microcontroller Unit	微控制单元
O		
OS	Operating System	操作系统
P		
PCIe	Peripheral Component Interconnect Express	快捷外围部件互连标准
S		
SMbus	System Management Bus	系统管理总线

A.3 kubelet 代码实现

说明

func (k KubeletStub) Register(ctx context.Context, r *pluginapi.RegisterRequest) (*pluginapi.Empty, error) {

kubelet实现代码中，如上述函数应定义为一行，由于PDF文档格式限制，超出单行部分可能会自动换行，导致编译失败，此时需要您手动将换行的部分与前一行合并为一行。

```
// k8s test client
package main

import (
    "flag"
    "fmt"
    "io/ioutil"
    "log"
    "net"
    "os"
    "os/signal"
    "path"
    "sync"
    "syscall"
    "time"

    "golang.org/x/net/context"
    "google.golang.org/grpc"

    pluginapi "k8s.io/kubernetes/pkg/kubelet/apis/deviceplugin/v1beta1"
)

const (
    debug = false
    sleepTime = 5
    grpcTimeOut = 10
    testNum = 4
)

// KubeletStub for test
type KubeletStub struct {
    sync.Mutex
    server         *grpc.Server
    socket         string
    pluginEndpoint string
    testDir        string
}

// NewKubeletStub returns an initialized KubeletStub for testing purpose.
func NewKubeletStub(socket string) *KubeletStub {
    return &KubeletStub{
        socket: socket,
    }
}

// Register from Device Plugin
func (k KubeletStub) Register(ctx context.Context, r *pluginapi.RegisterRequest) (*pluginapi.Empty, error) {
    k.Lock()
    defer k.Unlock()
    k.pluginEndpoint = r.Endpoint
    go testTask(k.testDir, k.pluginEndpoint)
    return &pluginapi.Empty{}, nil
}

// Start Server and Register to Deivce Plugin
func (k *KubeletStub) Start() error {
    os.Remove(k.socket)
```

```
    s, err := net.Listen("unix", k.socket)
    if err != nil {
        fmt.Printf("Can't listen at the socket: %+v", err)
        return err
    }

    k.server = grpc.NewServer([]grpc.ServerOption{}...)

    pluginapi.RegisterRegistrationServer(k.server, k)
    go k.server.Serve(s)
    return nil
}

func main() {
    flag.Parse()

    testDir := os.TempDir() + "/hiai_plugin_test"
    if _, err := os.Stat(testDir); err != nil {
        log.Printf("%s not exist\n", testDir)
        return
    }
    log.Println(testDir)

    log.Println("Starting OS watcher.")
    sigs := newOSWatcher(syscall.SIGHUP, syscall.SIGINT, syscall.SIGTERM, syscall.SIGQUIT)

    kubeletEndpoint := path.Join(testDir, "kubelet.sock")
    kubeletStub := NewKubeletStub(kubeletEndpoint)
    kubeletStub.testDir = testDir
    err := kubeletStub.Start()
    if err != nil {
        log.Println("KubeletStub Server Start Error")
        return
    }
    defer kubeletStub.server.Stop()
    log.Println("KubeletStub Server Start")

    time.Sleep(sleepTime * time.Second)

    select {
    case s := <-sigs:
        switch s {
        case syscall.SIGHUP:
            log.Println("Received SIGHUP, restarting.")
            removeAllSocks(testDir)
            os.Remove(kubeletEndpoint)
        default:
            log.Printf("Received signal \"%v\", shutting down.", s)
            os.Remove(kubeletEndpoint)
            removeAllSocks(testDir)
            break
        }
    }
}

func newOSWatcher(sigs ...os.Signal) chan os.Signal {
    sigChan := make(chan os.Signal, 1)
    signal.Notify(sigChan, sigs...)

    return sigChan
}

// Remove all sock file on the local node by walking `/testDir` directory.
func removeAllSocks(testDir string) error {
    files, err := ioutil.ReadDir(testDir)
    if err != nil {
        return err
    }
    for _, f := range files {
```

```
        if f.IsDir() {
            continue
        }
        os.Remove(testDir + "/" + f.Name())
    }
    return nil
}

func testTask(testDir, pluginEndpoint string) {
    devicePluginSock := path.Join(testDir, pluginEndpoint)
    // Verifies the grpcServer is ready to serve services.
    conn, err := grpc.Dial(devicePluginSock, grpc.WithInsecure(), grpc.WithBlock(),
        grpc.WithTimeout(grpcTimeOut * time.Second),
        grpc.WithDialer(func(addr string, timeout time.Duration) (net.Conn, error) {
            return net.DialTimeout("unix", addr, timeout)
        }))
    if err != nil {
        log.Printf("Connect device plugin fail, %+v\n", err)
        return
    }

    client := pluginapi.NewDevicePluginClient(conn)

    // Tests ListAndWatch
    fmt.Printf("\n---> ListAndWatch\n")
    stream, err := client.ListAndWatch(context.Background(), &pluginapi.Empty{})
    devs, err := stream.Recv()
    fmt.Printf("%+v\n\n", devs)

    time.Sleep(1 * time.Second)

    devices := make([]pluginapi.Device, 0)
    for _, d := range devs.Devices {
        dev := pluginapi.Device{
            ID:     d.ID,
            Health: d.Health,
        }
        devices = append(devices, dev)
    }

    allocateID := make([]string, 0)
    if len(devices) > testNum {
        allocateID = append(allocateID, devices[0].ID, devices[1].ID, devices[2].ID)
    } else if len(devices) == testNum {
        allocateID = append(allocateID, devices[0].ID, devices[1].ID)
    } else {
        allocateID = append(allocateID, devices[0].ID)
    }

    // Tests Allocate
    fmt.Printf("\n---> Allocate\n")
    resp, err := client.Allocate(context.Background(), &pluginapi.AllocateRequest{
        ContainerRequests: []*pluginapi.ContainerAllocateRequest{
            {DevicesIDs: allocateID}}})
    fmt.Printf("%+v\n\n", resp)

    time.Sleep(1 * time.Second)

    conn.Close()
}
```

A.4 免责声明

- 本文档可能包含第三方信息、产品、服务、软件、组件、数据或内容（统称“第三方内容”）。华为不控制且不对第三方内容承担任何责任，包括但不限于准确性、兼容性、可靠性、可用性、合法性、适当性、性能、不侵权、更新状态等，

除非本文档另有明确说明。在本文档中提及或引用任何第三方内容不代表华为对第三方内容的认可或保证。

- 用户若需要第三方许可，须通过合法途径获取第三方许可，除非本文档另有明确说明。

A.5 如何获取帮助

日常维护或故障处理过程中遇到难以解决或者重大问题时，请寻求华为技术有限公司的技术支持。

A.5.1 收集必要的故障信息

在进行故障处理前，需要收集必要的故障信息。

收集的信息主要包括：

- 客户的详细名称、地址
- 联系人姓名、电话号码
- 故障发生的具体时间
- 故障现象的详细描述
- 设备类型及软件版本
- 故障后已采取的措施和结果
- 问题的级别及希望解决的时间

A.5.2 做好必要的调试准备

在寻求华为技术支持时，华为技术支持工程师可能会协助您做一些操作，以进一步收集故障信息或者直接排除故障。

在寻求技术支持前请准备好单板和端口模块的备件、螺丝刀、螺丝、串口线、网线等可能使用到的物品。

A.5.3 如何使用文档

华为技术有限公司提供全面的随设备发货的指导文档。指导文档能解决您在日常维护或故障处理过程中遇到的常见问题。

为了更好的解决故障，在寻求华为技术支持前，建议充分使用指导文档。

A.5.4 获取技术支持

华为技术有限公司通过办事处、公司二级技术支持体系、电话技术指导、远程支持及现场技术支持等方式向用户提供及时有效的技术支持。

技术支持网址

查阅技术资料合集：**https://e.huawei.com/cn/** > 技术支持 > 产品和解决方案支持 > 服务器-智能计算 > 昇腾计算

查阅技术资料的使用流程：**https://www.hiascend.com** > 文档

自助平台与论坛

如果您想进一步学习和交流：

- 访问华为服务器信息服务平台，获取相关服务器产品资料。
- 访问华为企业业务智能问答系统，快速查询产品问题。
- 访问华为企业互动社区（服务器），进行硬件产品学习交流。
- 访问开发者论坛，进行AI应用开发学习交流。

公告

有关产品生命周期、预警和整改公告请访问技术支持 > 公告 > 产品公告。

案例库

参阅已有案例进行学习：计算产品案例查询助手。

说明

计算产品案例查询助手目前仅面向华为合作伙伴及华为工程师开放。

获取华为技术支持

如果在设备维护或故障处理过程中，遇到难以确定或难以解决的问题，通过文档的指导仍然不能解决，请通过如下方式获取技术支持：

- 联系华为技术有限公司客户服务中心。

  中国区企业用户请通过以下方式联系我们：
  - 客户服务电话：400-822-9999
  - 客户服务邮箱：**support_e@huawei.com**

    企业网全球各地区客户服务热线可以通过以下网站查找：企业用户全球服务热线

  中国区运营商用户请通过以下方式联系我们：
  - 客户服务电话：400-830-2118
  - 客户服务邮箱：**support@huawei.com**

    运营商全球各地区客户服务热线可以通过以下网站查找：运营商用户全球服务热线
- 联系华为技术有限公司驻当地办事处的技术支持人员。