

IN200 网卡

# 用户指南

文档版本 19

发布日期 2024-11-15



版权所有 © 华为技术有限公司 2024。保留一切权利。

未经本公司书面许可，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://www.huawei.com>

客户服务邮箱：[support@huawei.com](mailto:support@huawei.com)

客户服务电话：4008302118

# 安全声明

## 漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

# 前言

## 概述

本文档详细的描述了IN200标准PCIe网卡的功能特性、软件升级方法，以及所需工具的安装和使用方法。

## 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
	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。
	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。
	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。
	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。
	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。

## 修改记录

修改记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

文档版本	发布日期	修改说明
19	2024-11-15	第十九次正式发布。 文档名称由“华为 IN200 网卡 用户指南”修改为“IN200 网卡 用户指南”。
18	2024-08-09	第十八次正式发布。 修改 <b>1.4 技术规格</b> 中“PCIe接口”，新增说明。
17	2021-10-18	第十七次正式发布。 删除支持IPv4/IPv6相关描述。
16	2021-07-20	第十六次正式发布。 <ul style="list-style-type: none"><li>● 增加<b>3.1.2.2 在产品路径下下载软件包</b>。</li><li>● 修改<b>3.2.2 RoCE驱动维护</b>。</li><li>● 增加<b>4.4.2 查询BMC与IN200的带外交互信息 ( bmc_show )</b>。</li><li>● 增加<b>6.5.2 配置2*100GE网卡64队列</b>。</li></ul>
15	2021-05-26	第十五次正式发布。 <ul style="list-style-type: none"><li>● 修改<b>1.1 概述</b>中端口描述。</li><li>● 修改<b>1.3 特性</b>中“自协商”相关描述。</li></ul>
14	2021-04-12	第十四次正式发布。 <ul style="list-style-type: none"><li>● 修改涉及RoCE驱动支持的OS列表刷新。</li><li>● 修改<b>3.3.2 固件升级</b>：补充说明固件升级的方式。</li><li>● 修改<b>6.4.3.1 通过命令行配置RoCE bonding</b>。</li><li>● 增加7个hinicadm命令。 ( <b>currenet_port_traffic</b>、 <b>currenet_cc_traffic</b>、 <b>mig_dirty_bitmap</b>、<b>mig_info</b>、 <b>mig_queue</b>、<b>mig_vf</b>和<b>xspf</b> )</li></ul>
13	2021-03-11	第十三次正式发布。 修改 <b>6.5.1.2 配置方法</b> 。

文档版本	发布日期	修改说明
12	2021-01-22	<p>第十二次正式发布。</p> <ul style="list-style-type: none"><li>增加<a href="#">3.1.2.1 在Computing Component iDriver下下载软件包</a>。</li><li>增加<a href="#">4.4.20 查询和设置链路跟随状态 ( hilink_follow )</a>。</li><li>增加<a href="#">4.4.34 查询OVS卸载信息 ( ovs_info )</a>。</li><li><b>6.5.1.1 概述</b>: 补充说明“当前VF队列数为4个时，单VF无法达到100GE卡极限带宽。”</li></ul>
11	2020-12-25	<p>第十一次正式发布。</p> <ul style="list-style-type: none"><li>修改“系统要求”改为“<a href="#">2 兼容性</a>”。</li><li>合入SP586、SC371和SC381相关内容。</li><li>修改<a href="#">3.1.1 获取软件包 ( SPC501及以下版本)</a>。</li><li>增加<a href="#">4.4.16 查询和设置过滤功能 ( filter )</a>。</li></ul>
10	2020-05-29	<p>第十次正式发布。</p> <ul style="list-style-type: none"><li>修改系统要求。</li><li>增加<a href="#">6.7.2 在ARM架构下常用BIOS配置</a>。</li><li>增加<a href="#">1.3 特性</a>增加支持特性：“虚拟机多队列”和“DPDK”。</li></ul>
09	2020-02-21	<p>第九次正式发布。</p> <p>删除Windows系统相关的规格、驱动安装/升级/卸载等内容。</p>
08	2020-01-10	<p>第八次正式发布。</p> <ul style="list-style-type: none"><li>修改系统要求。</li><li>增加<a href="#">4.4.43.10 查询SDI卡服务器侧重启, 芯片是否跟随复位 ( -t reset_scheme )</a>。</li><li>增加<a href="#">4.4.43.11 设置SDI卡服务器侧重启, 芯片是否跟随复位 ( -t reset_scheme -e )</a>。</li></ul>
07	2019-11-15	<p>第七次正式发布。</p> <ul style="list-style-type: none"><li>修改<a href="#">1 认识IN200</a>。</li><li>增加<a href="#">6.5.1 配置4*25GE网卡64队列</a>。</li><li>增加<a href="#">6.6 日志收集</a>。</li></ul>

文档版本	发布日期	修改说明
06	2019-07-27	第六次正式发布。 修改系统要求。
05	2019-07-15	第五次正式发布。 <ul style="list-style-type: none"><li>• 修改<a href="#">1.3 特性</a>。</li><li>• 修改系统要求。</li><li>• 修改<a href="#">3.2.1 NIC驱动维护</a>中刷新安装和升级NIC驱动的实例。</li><li>• 修改<a href="#">3.2.2 RoCE驱动维护</a>中增加安装OFED的相关操作。</li><li>• 新增<a href="#">6.3 配置QoS</a>。</li><li>• 新增<a href="#">6.4 配置RoCE bonding</a>。</li><li>• 新增<a href="#">4.4.39 设置和触发网卡在AC掉电时待发送的报文数据(reboot_notice)</a> ~<a href="#">4.4.12.3 设置DWRR调度(-t -p)</a>。</li></ul>
04	2019-05-10	第四次正式发布。 <ul style="list-style-type: none"><li>• 新增<a href="#">3.2.1 NIC驱动维护</a>中增加Windows系统下驱动维护相关内容。</li><li>• 新增<a href="#">4 hinicadm管理工具</a>中增加Windows系统下hinicadm工具的安装、升级和卸载等相关内容。</li><li>• 新增<a href="#">3.2.2 RoCE驱动维护</a>。</li><li>• 新增<a href="#">6.2 配置SR-IOV</a>。</li></ul>
03	2019-01-30	第三次正式发布。
02	2018-08-10	第二次正式发布。
01	2018-05-20	第一次正式发布。

# 目 录

<b>前言</b>	iii
<b>1 认识 IN200</b>	1
1.1 概述	1
1.2 外观结构	3
1.3 特性	11
1.4 技术规格	13
<b>2 兼容性</b>	15
<b>3 安装与维护</b>	16
3.1 获取软件包	16
3.1.1 获取软件包（SPC501 及以下版本）	16
3.1.2 获取软件包（SPC602 及以上版本）	18
3.1.2.1 在 Computing Component iDriver 下下载软件包	18
3.1.2.2 在产品路径下下载软件包	21
3.1.3 软件完整性验证	23
3.2 管理驱动	23
3.2.1 NIC 驱动维护	23
3.2.1.1 使用一键脚本安装升级驱动	23
3.2.1.2 准备工作（SUSE）	25
3.2.1.3 安装驱动	26
3.2.1.4 升级驱动	28
3.2.1.5 卸载驱动	30
3.2.2 RoCE 驱动维护	31
3.2.2.1 RoCE 支持的操作系统	31
3.2.2.2 安装驱动	32
3.2.2.3 升级驱动	35
3.2.2.4 卸载驱动	38
3.3 管理固件	39
3.3.1 使用一键脚本升级固件	40
3.3.2 固件升级	41
<b>4 hinicadm 管理工具</b>	43
4.1 安装 hinicadm	43
4.2 使用方法	44

4.3 命令列表.....	44
4.4 命令介绍.....	50
4.4.1 查询和设置指定设备的自协商模式 ( autoneg ) .....	50
4.4.2 查询 BMC 与 IN200 的带外交互信息 ( bmc_show ) .....	51
4.4.3 查询芯片反压信息 ( bp ) .....	54
4.4.4 清除指定设备统计信息 ( clear ) .....	55
4.4.5 查询统计信息 ( counter ) .....	56
4.4.5.1 查询指定设备统计信息 ( -t -x ) .....	56
4.4.5.2 离线解析巡检信息 ( -o ) .....	57
4.4.6 查询芯片 CPB Cell 使用情况 ( cpb ) .....	58
4.4.7 查询芯片内部模块信息并保存 ( csr_info ) .....	58
4.4.8 在 currenet 场景下查询 RoCE 各端口发送接收实时流量 ( currenet_port_traffic ) .....	59
4.4.9 在 currenet 场景下查询和设置拥塞控制算法的参数 ( currenet_cc_param ) .....	60
4.4.10 查询和设置 DCB 功能 ( dcb ) .....	61
4.4.11 查询芯片数据通路相关模块统计信息 ( dp ) .....	61
4.4.12 配置 ETS 功能 ( ets ) .....	62
4.4.12.1 查询和设置 ETS 功能 ( -e ) .....	62
4.4.12.2 设置 SP 调度 ( -t ) .....	63
4.4.12.3 设置 DWRR 调度 ( -t -p ) .....	64
4.4.13 查询异步事件统计信息 ( event ) .....	65
4.4.14 查询异常记录 ( fe_epc ) .....	66
4.4.14.1 获取指定设备的异常信息 ( show ) .....	66
4.4.14.2 触发指定 FE 产生并记录异常信息 ( -c -t ) .....	67
4.4.15 查询和设置指定端口 FEC 模式 ( fec ) .....	68
4.4.15.1 查询端口 FEC 模式 ( -p ) .....	68
4.4.15.2 设置指定端口 FEC 模式 ( -p -m ) .....	68
4.4.15.3 清除现有配置 ( -p -c ) .....	69
4.4.16 查询和设置过滤功能 ( filter ) .....	70
4.4.17 查询硬件模块上报的中断信息 ( fm_show ) .....	71
4.4.18 查询指定端口的统计信息 ( hilink_count ) .....	71
4.4.19 查询自协商训练寄存器信息 ( hilink_dump ) .....	72
4.4.20 查询和设置链路跟随状态 ( hilink_follow ) .....	73
4.4.21 查询 hi16/hi30 参数 ( hilink_param ) .....	73
4.4.22 查询指定端口的基本信息 ( hilink_port ) .....	74
4.4.23 查询和设置端口速率参数 ( hilink_speed ) .....	75
4.4.23.1 查询指定端口的当前速率 ( -p ) .....	76
4.4.23.2 设置指定端口的当前速率 ( -p -s ) .....	76
4.4.23.3 设置指定端口的自协商模式的开启状态 ( -p -an ) .....	77
4.4.23.4 设置指定端口的连接模式 ( -p -m ) .....	78
4.4.23.5 清除指定端口的持久化速率配置 ( -p -c ) .....	79
4.4.24 查询 IN200 基本信息 ( info ) .....	80
4.4.25 收集解析日志信息 ( log ) .....	81

4.4.25.1 收集指定设备的所有固件日志信息 ( -a ) .....	81
4.4.25.2 收集指定类型的日志信息 ( -t ) .....	82
4.4.25.3 离线解析运行日志信息 ( -o1 ) .....	82
4.4.25.4 离线解析临终遗言 ( -o2 ) .....	83
4.4.26 查询和设置 LRO 参数 ( lro ) .....	84
4.4.26.1 查询整卡的 LRO 参数.....	84
4.4.26.2 统一设置整卡所有 function 的 LRO 聚合个数 ( -n ) .....	85
4.4.26.3 设置整卡的 LRO 聚合时间 ( -t ) .....	85
4.4.26.4 查询单个 function 的 LRO 参数 ( -x ) .....	86
4.4.26.5 设置单个 function 的 LRO 聚合个数 ( -x -n ) .....	87
4.4.27 查询 MAC 地址 ( mac ) .....	87
4.4.28 查询热迁移指定虚拟机 ID 的脏页信息 ( mig_dirty_bitmap ) .....	88
4.4.29 查询热迁移驱动的基本信息 ( mig_info ) .....	89
4.4.30 查询热迁移驱动的队列信息 ( mig_queue ) .....	90
4.4.31 查询热迁移驱动 vf 的基本信息或者统计参数 ( mig_vf ) .....	91
4.4.32 查询和设置指定设备的工作模式 ( mode ) .....	92
4.4.33 查询指定设备队列信息 ( nic_queue ) .....	93
4.4.34 查询 OVS 卸载信息 ( ovs_info ) .....	94
4.4.35 查询 CPB Cell 资源分配以及当前资源占用情况 ( pdm ) .....	94
4.4.35.1 查询某个模块下的 CPB Cell 资源分配以及当前资源占用情况 ( -m ) .....	95
4.4.35.2 查询某个 index 下的 CPB Cell 资源分配以及当前资源占用情况 ( -m -x ) .....	95
4.4.35.3 查询某段范围下的 CPB Cell 资源分配以及当前资源占用情况 ( -m -s -e ) .....	96
4.4.36 查询和设置 PFC 功能 ( pfc ) .....	97
4.4.36.1 查询 PFC 功能.....	97
4.4.36.2 设置 PFC 功能 ( -e -p ) .....	97
4.4.37 查询和设置用户优先级、cqe 水线 ( qos ) .....	98
4.4.38 查询和设置端口限速速率 ( rate ) .....	99
4.4.39 设置和触发网卡在 AC 掉电时待发送的报文数据 ( reboot_notice ) .....	100
4.4.40 恢复出厂设置 ( reset ) .....	101
4.4.41 配置 ROCE_DSCP 功能 ( roce_dscp ) .....	102
4.4.41.1 查询 ROCE_DSCP 功能.....	102
4.4.41.2 设置 ROCE_DSCP 功能 ( -e ) .....	103
4.4.41.3 重置 ROCE_DSCP 功能的优先级映射规则 ( -r ) .....	103
4.4.41.4 设置 DSCP 与用户优先级的关系 ( -s -p ) .....	104
4.4.41.5 删除 DSCP 与用户优先级的映射关系 ( -d -p ) .....	105
4.4.42 查询 RoCE 各端口发送接收实时流量 ( roce_port_traffic ) .....	106
4.4.43 查询和设置 SDI 配置 ( sdi_cfg ) .....	107
4.4.43.1 查询配置的管理 VLAN ( -t cpath_vlan ) .....	107
4.4.43.2 配置管理 VLAN ( -t cpath_vlan -v -s ) .....	107
4.4.43.3 删除配置的管理 VLAN ( -t cpath_vlan -v -c ) .....	108
4.4.43.4 查询指定 PF 发送方向限速参数 ( -t pf_tx_rate ) .....	109
4.4.43.5 设置指定 PF 发送方向限速参数 ( -t pf_tx_rate -l ) .....	109

4.4.43.6 删除指定 PF 发送方向限速参数 ( -t pf_tx_rate -c ) .....	110
4.4.43.7 查询指定 PF 接收方向限速参数 ( -t pf_rx_rate ) .....	110
4.4.43.8 设置指定 PF 接收方向限速参数 ( -t pf_rx_rate -l ) .....	111
4.4.43.9 删除指定 PF 接收方向限速参数 ( -t pf_rx_rate -c ) .....	112
4.4.43.10 查询 SDI 卡服务器侧重启, 芯片是否跟随复位 ( -t reset_scheme ) .....	112
4.4.43.11 设置 SDI 卡服务器侧重启, 芯片是否跟随复位 ( -t reset_scheme -e ) .....	113
4.4.43.12 查询 SDI 卡上服务器侧 PT 设备的显示状态 ( -t set_pt ) .....	113
4.4.43.13 设置 SDI 卡服务器侧 PT 设备的显示状态 ( -t set_pt -d ) .....	114
4.4.44 查询和设置 SDI 卡模式 ( sdi_mode ) .....	115
4.4.44.1 查询 SDI 卡模式.....	115
4.4.44.2 设置 SDI 卡模式 ( -m ) .....	115
4.4.45 查询和设置端口协商自适应状态 ( self_adaption ) .....	116
4.4.45.1 查询端口协商自适应状态 ( -p ) .....	116
4.4.45.2 清除配置 ( -p -c ) .....	117
4.4.45.3 设置端口协商自适应状态 ( -p -s -t -d ) .....	117
4.4.46 查询指定配置类型的 serdes 信息 ( serdes ) .....	118
4.4.47 查询指定端口的光模块信息 ( sfp ) .....	119
4.4.48 查询和设置指定端口的虚拟化控制模式的启用状态 ( sriov ) .....	120
4.4.49 查询指定设备的线性表 ( table ) .....	121
4.4.50 查询芯片温度和光模块温度 ( temperature ) .....	123
4.4.51 查询微码线程状态 ( tile_io ) .....	123
4.4.52 升级固件 ( updatefw ) .....	124
4.4.52.1 冷升级整体固件 ( -f ) .....	124
4.4.52.2 冷升级 FW 固件及配置文件 ( -f -c ) .....	125
4.4.52.3 热升级固件 ( -f -a ) .....	126
4.4.52.4 查询当前 IN200 的固件能否被指定版本热升级 ( -v -t ) .....	127
4.4.53 查询设备版本信息 ( version ) .....	127
4.4.54 管理 SFP/QSFP 的信息和状态 ( xsfp ) .....	128
4.5 升级 hinicadm.....	130
4.6 卸载 hinicadm.....	131
<b>5 标准管理工具.....</b>	<b>132</b>
<b>6 常用配置.....</b>	<b>136</b>
6.1 配置驱动加载参数.....	136
6.1.1 驱动加载参数说明.....	136
6.1.2 配置链路状态跟随的加载参数.....	138
6.2 配置 SR-IOV.....	140
6.2.1 X86 版本下配置 SR-IOV.....	140
6.2.1.1 配置 BIOS.....	140
6.2.1.2 修改系统引导配置文件.....	147
6.2.2 ARM 版本下配置 SR-IOV.....	148
6.2.2.1 配置 BIOS.....	148
6.2.2.2 修改系统引导配置文件.....	151

6.3 配置 QoS.....	152
6.3.1 支持流控概述.....	152
6.3.2 基于 VLAN tag 的流控配置方法.....	152
6.3.3 基于 DSCP 的流控配置方法.....	153
6.4 配置 RoCE bonding.....	153
6.4.1 RoCE bonding 概述.....	153
6.4.2 RoCE bonding 支持的操作系统.....	154
6.4.3 配置方法.....	155
6.4.3.1 通过命令行配置 RoCE bonding.....	155
6.4.3.2 通过配置文件配置 RoCE bonding.....	157
6.5 配置 64 队列.....	158
6.5.1 配置 4*25GE 网卡 64 队列.....	158
6.5.1.1 概述.....	158
6.5.1.2 配置方法.....	159
6.5.1.3 恢复默认配置.....	160
6.5.2 配置 2*100GE 网卡 64 队列.....	162
6.5.2.1 概述.....	162
6.5.2.2 配置方法.....	162
6.5.2.3 恢复默认配置.....	163
6.6 日志收集.....	165
6.6.1 驱动和固件日志收集.....	165
6.6.1.1 安装日志收集脚本.....	165
6.6.1.2 运行日志收集脚本.....	165
6.6.1.3 日志文件清单.....	166
6.6.2 PXE 日志收集.....	169
6.7 常用 BIOS 配置.....	170
6.7.1 在 x86 架构下常用 BIOS 配置.....	170
6.7.2 在 ARM 架构下常用 BIOS 配置.....	176
<b>A FAQ.....</b>	<b>183</b>
A.1 IN200 网卡 VF 进行 FLR 时概率性出现虚拟机操作系统加载驱动失败的处理方法.....	183
<b>B 通过虚拟文件夹传输文件.....</b>	<b>184</b>
<b>C 缩略语.....</b>	<b>185</b>

# 1 认识 IN200

- [1.1 概述](#)
- [1.2 外观结构](#)
- [1.3 特性](#)
- [1.4 技术规格](#)

## 1.1 概述

IN200以太网卡（以下简称IN200）是适配华为服务器的PCIe标卡/OCP3.0网卡，为服务器提供扩展的对外业务接口。

IN200网卡根据接口数量和速率不同有以下几种：

网卡名称	接口描述
SP570和SP580	提供4个25GE/10GE SFP28光口
SP572和SC371	提供2个100GE/40GE QSFP28光口
SP582、SP586和SC381	提供2个25GE/10GE SFP28光口

IN200是基于华为海思Hi1822网卡芯片的PCIe标卡/OCP3.0网卡，支持PCIe x16 Gen3.0（SP570/SP580/SP572/SC371），PCIe x8 Gen3.0（SP582/SP586/SC381），支持I<sup>2</sup>C（Inter-integrated Circuit）和MCTP（Management Component Transport Protocol）带外管理。

IN200的基本架构如图1-1～图1-3所示。

图 1-1 SP570/SP580 网卡基本架构

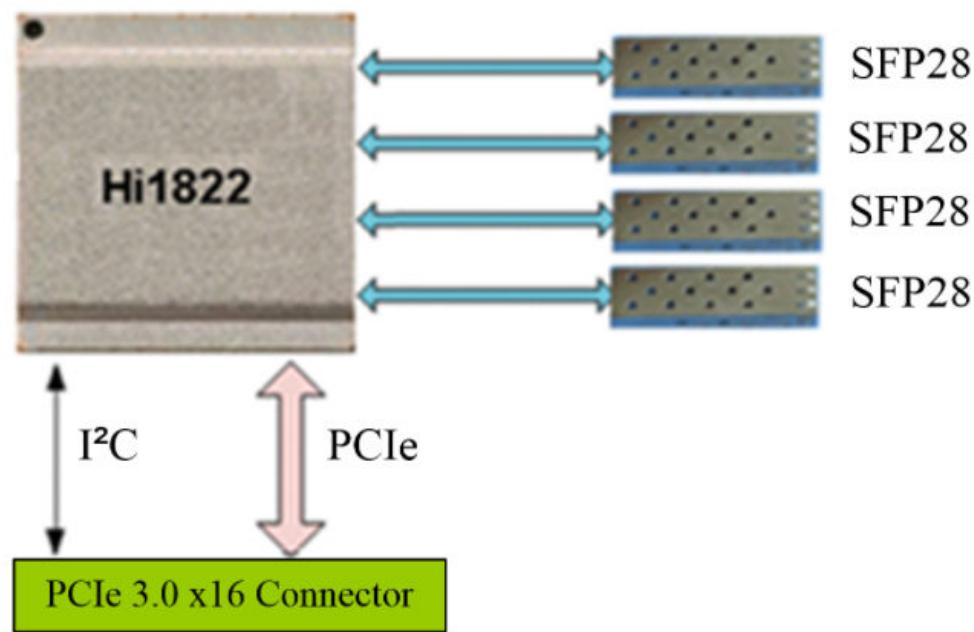


图 1-2 SP572/SC371 网卡基本架构

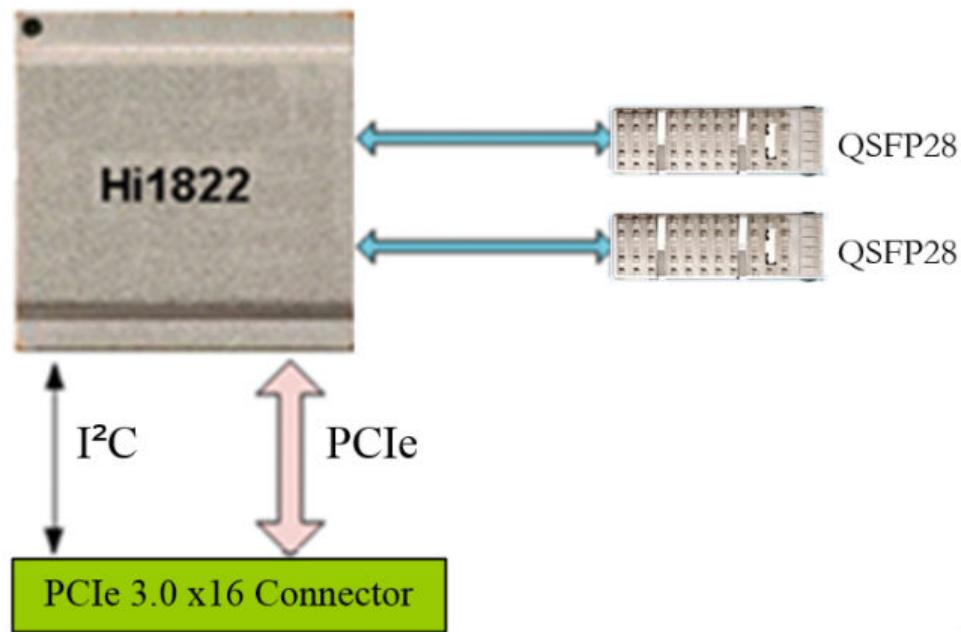
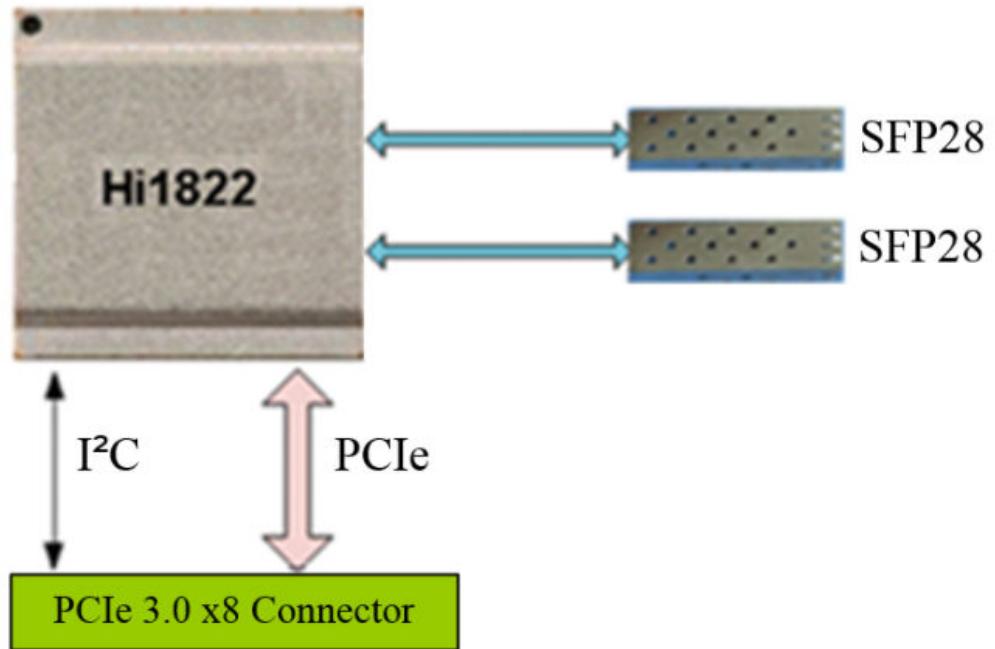


图 1-3 SP582/SP586/SC381 网卡基本架构



## 1.2 外观结构

### 部件

IN200的主要部件如[图1-4 ~ 图1-10](#)所示。

图 1-4 SP570 部件

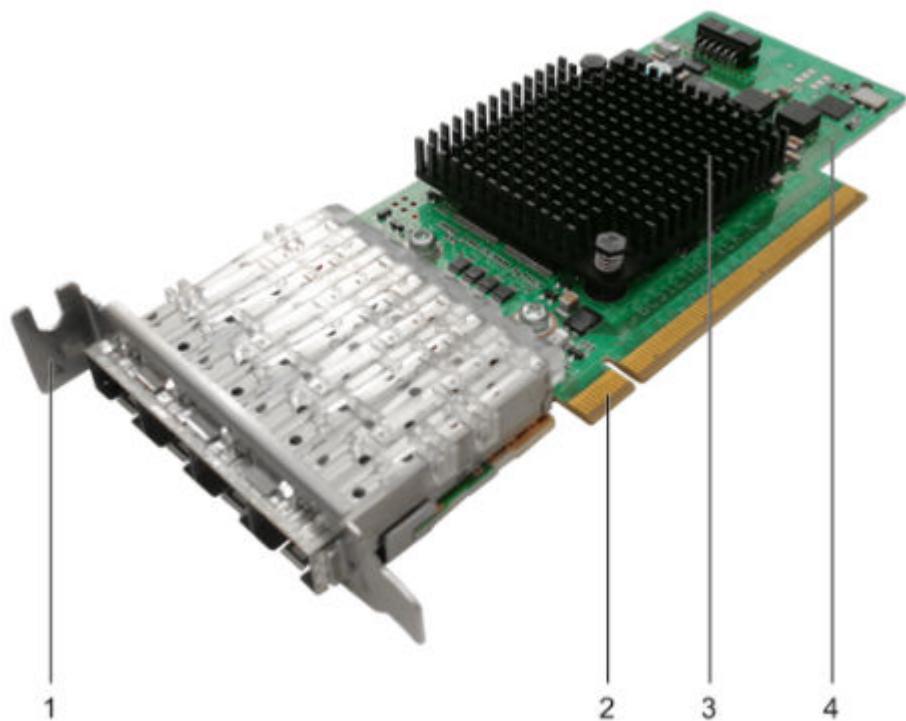


图 1-5 SP580 部件

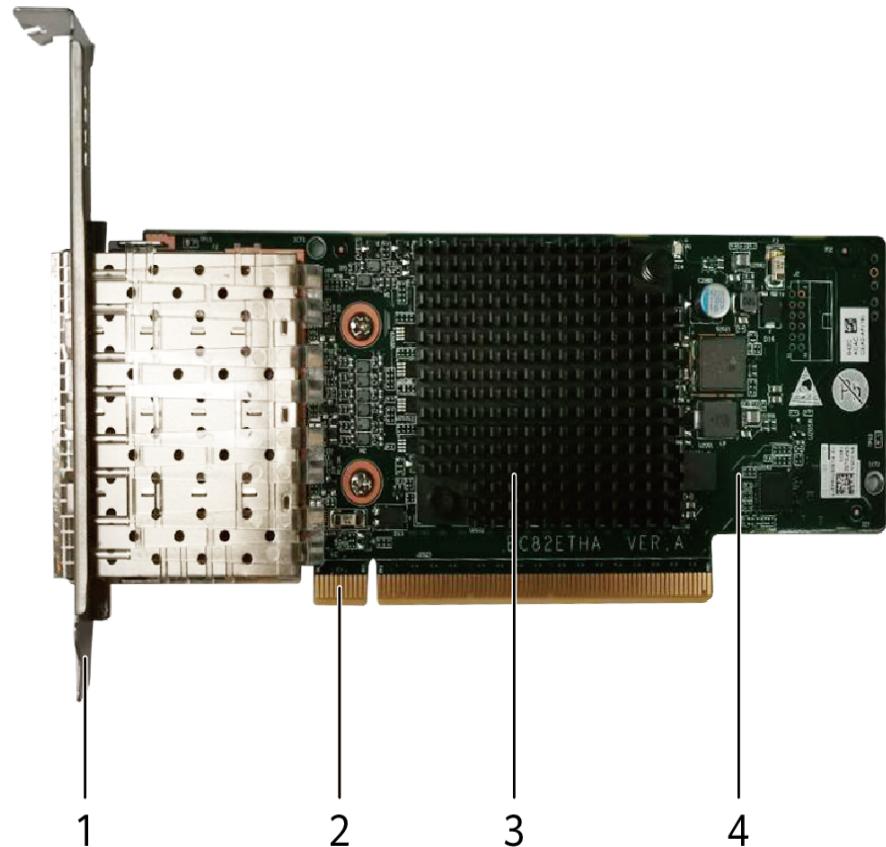


图 1-6 SP572 部件

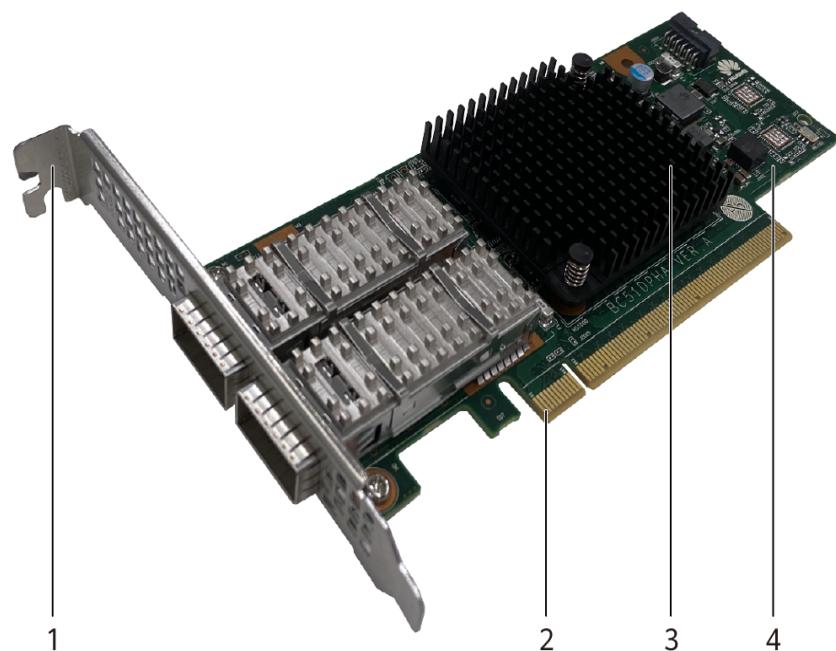


图 1-7 SP582 部件

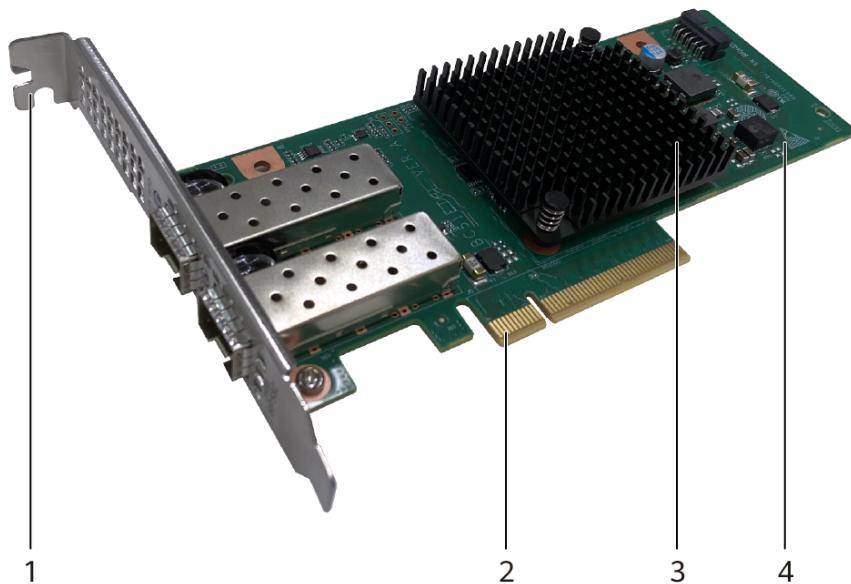


图 1-8 SP586 部件

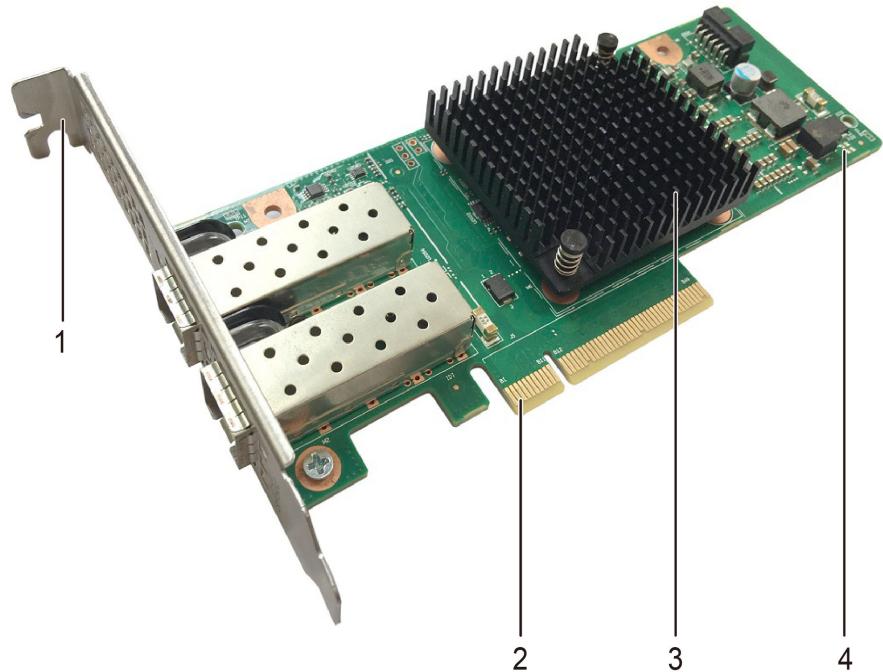


表 1-1 SP570/SP580/SP572/SP582/SP586 部件

1	拉手条（支持全高和半高拉手条，具体请以实际为准）	2	PCIe连接器
3	Hi1822芯片+散热器 <b>说明</b> Hi1822芯片位于散热器的下方。	4	主板

图 1-9 SC371 部件

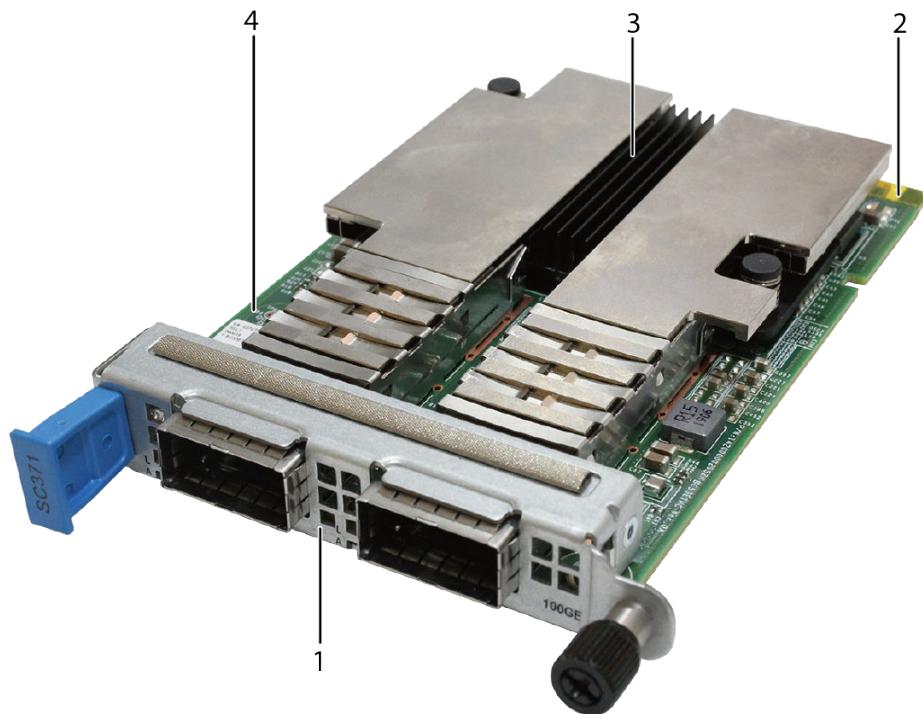


图 1-10 SC381 部件

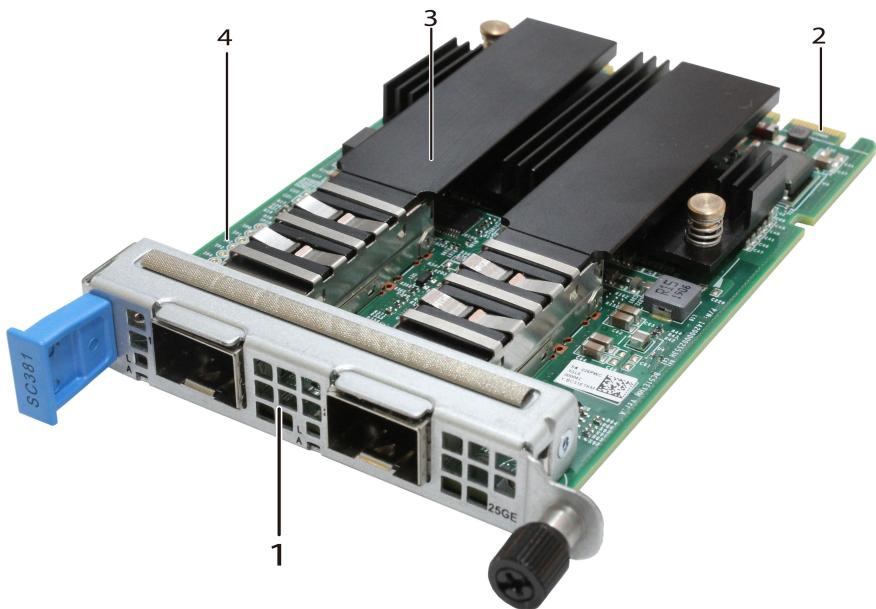


表 1-2 SC371/SC381 部件

1	拉手条	2	OCP连接器
3	Hi1822+散热器 <b>说明</b> Hi1822芯片位于散热器的下方。	4	主板

IN200各个部件的功能如表1-3所示。

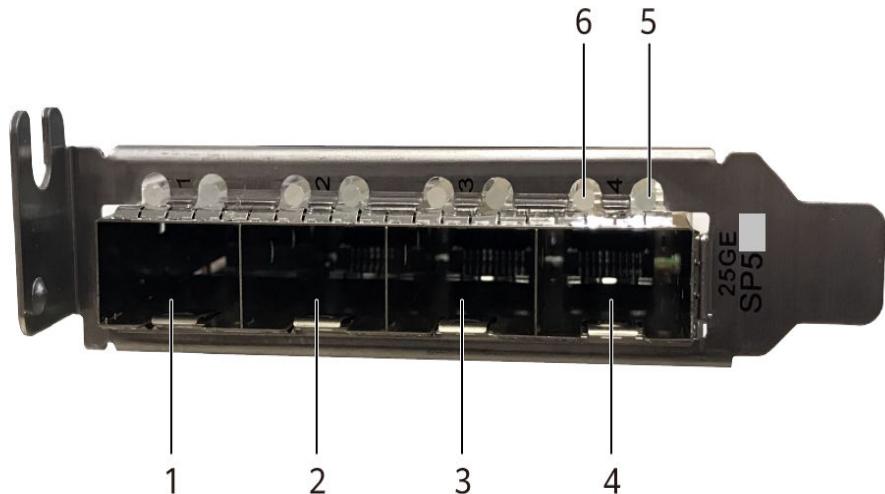
表 1-3 IN200 部件说明

名称	说明
主板	包含网卡模块、网络接口模块、电源模块等。
Hi1822	网卡主芯片，实现网卡功能。
散热器	用于网卡芯片散热。
拉手条	网卡拉手条，支持全高和半高拉手条。
PCIe连接器/OCP连接器	连接服务器PCIe插槽。

## 面板

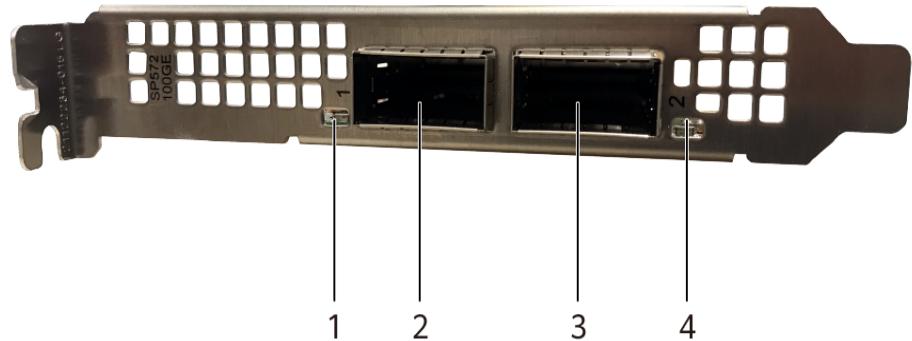
IN200的面板及接口如图1-11 ~ 图1-15所示。

图 1-11 SP570/SP580 面板



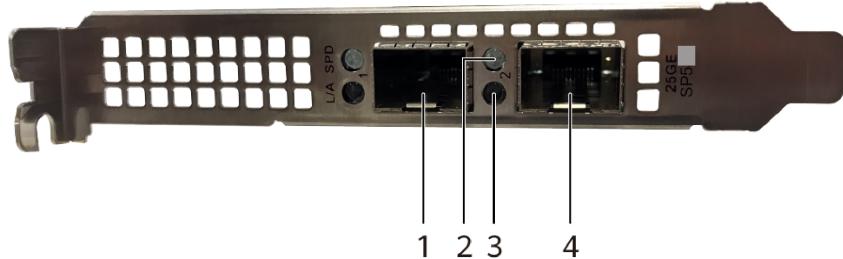
1	SFP28光口1	2	SFP28光口2
3	SFP28光口3	4	SFP28光口4
5	Active/Link指示灯	6	Speed指示灯

图 1-12 SP572 面板



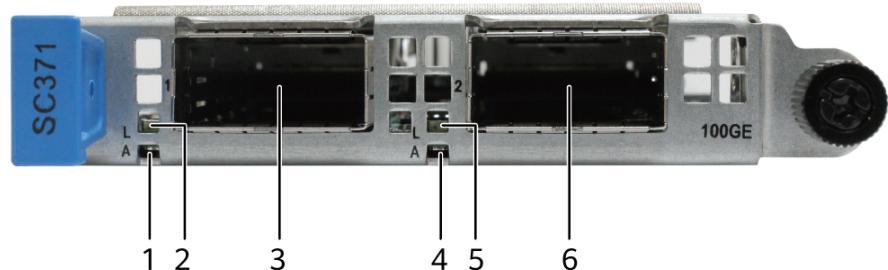
1/ 4	Active/Link指示灯+Speed指示灯	2	QSFP28光口1
3	QSFP28光口2	-	-

图 1-13 SP582/SP586 面板



1	SFP28光口1	2	Speed指示灯
3	Active/Link指示灯	4	SFP28光口2

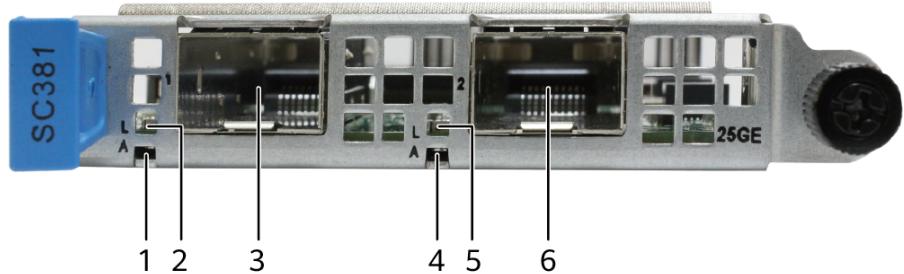
图 1-14 SC371 面板



1/4	Active指示灯	2/5	Link指示灯
-----	-----------	-----	---------

3	QSFP28光口1	6	QSFP28光口2
---	-----------	---	-----------

图 1-15 SC381 面板



1/4	Active指示灯	2/5	Link指示灯
3	SFP28光口1	6	SFP28光口2

## 指示灯

通过观察指示灯状态可以初步诊断当前IN200的状态。IN200面板上的指示灯的说明如表1-4 ~ 表1-7所示。

表 1-4 SP570/SP580/SP582/SP586 指示灯说明

指示灯名称	含义	颜色	说明
Active/Link 指示灯	网络连接 状态指示 灯	绿色	<ul style="list-style-type: none"><li>灭：链路没有Link。</li><li>常亮：链路有Link，无数据传输。</li><li>闪烁：链路有Link，有数据传输。</li></ul>
Speed指示 灯	网络数据 传输状态 指示灯	绿色/黄色	<ul style="list-style-type: none"><li>灭：链路没有Link。</li><li>黄常亮：链路有Link，速率10Gbps。</li><li>绿常亮：链路有Link，速率25Gbps。</li></ul>

表 1-5 SP572 指示灯说明

指示灯名称	含义	颜色	说明
Active/Link 指示灯 +Speed指 示灯	网络连接 状态指示 及数据传 输状态指 示灯	绿色+黄色	<ul style="list-style-type: none"><li>灭：链路没有Link。</li><li>绿常亮：链路有Link，无业务。</li><li>绿闪烁：链路有Link，有数据传输。分为两种情况：<ul style="list-style-type: none"><li>- 绿闪烁黄常亮：链路有Link，速率40Gbps。</li><li>- 绿闪烁黄不亮：链路有Link，速率100Gbps。</li></ul></li></ul>

表 1-6 SC371 指示灯说明

指示灯名称	含义	颜色	说明
Active指示 灯	网络连接 状态指示 灯	绿色	<ul style="list-style-type: none"><li>灭：链路没有Link。</li><li>常亮：链路有Link，无数据传输。</li><li>闪烁：链路有Link，有数据传输。</li></ul>
Link指示灯	网络数据 传输状态 指示灯	绿色/黄色	<ul style="list-style-type: none"><li>灭：链路没有Link。</li><li>黄常亮：链路有Link，速率100Gbps。</li><li>绿常亮：链路有Link，速率40Gbps。</li></ul>

表 1-7 SC381 指示灯说明

指示灯名称	含义	颜色	说明
Active指示 灯	网络连接 状态指示 灯	绿色	<ul style="list-style-type: none"><li>灭：链路没有Link。</li><li>常亮：链路有Link，无数据传输。</li><li>闪烁：链路有Link，有数据传输。</li></ul>
Link指示灯	网络数据 传输状态 指示灯	绿色/黄色	<ul style="list-style-type: none"><li>灭：链路没有Link。</li><li>黄常亮：链路有Link，速率25Gbps。</li><li>绿常亮：链路有Link，速率10Gbps。</li></ul>

## 1.3 特性

IN200的产品特性如下：

- SP570/SP580/SP572是半高半长PCIe x16标卡，SP582/SP586是半高半长PCIe x8标卡，SC371是PCIe x16 OCP3.0卡，SC381是PCIe x8 OCP3.0卡。

- SP570/SP580/SP572/SP582/SP586支持半高或全高拉手条，配置灵活，应用场景广泛。
- 采用华为海思Hi1822网卡专用芯片，SP570、SP580支持4个25GE SFP28端口。SP572/SC371支持2个100GE QSFP28端口。SP582/SP586/SC381支持2个25GE SFP28端口，与x86和ARM平台兼容性均良好。
- IN200支持在Legacy模式和UEFI ( Unified Extensible Firmware Interface ) 模式下使用。
- 支持100GE/40GE、25GE/10GE模式。当以太业务网口配置DAC电缆时，网卡可根据对端设备是否开启自协商来自调整业务网口是否开启自协商，即业务网口在自协商、RSFEC ( Reed-Solomon Forward Error Correction ) 、BASEFEC ( BASE Forward Error Correction ) 、NOFEC ( NO Forward Error Correction ) 模式之间轮询来与对端设备建立连接。
- 支持IPv4/IPv6/TCP/UDP校验和卸载，TSO ( TCP Segmentation Offload )，LRO ( Large Receive Offload ) 和RSS ( Receive Side Scaling )。
- 支持中断聚合参数配置和参数自适应。
- 支持802.1Q VLAN ( Virtual Local Area Network ) 加速和过滤。
- 支持VXLAN ( Virtual eXtensible Local Area Network ) 、NVGRE ( Network Virtualization using Generic Routing Encapsulation ) 卸载。
- 支持Pause帧、PFC ( Priority Flow Control ) 和ETS ( Enhanced Transmission Selection )。
- 支持NetQueue。
- 支持网卡SR-IOV ( Single Root I/O Virtualization )。
- 支持虚拟机多队列 ( Virtual Machine Multi Queue )。
- 支持DPDK ( Data Plane Development Kit )。
- 支持PF ( Physical Function ) 直通虚拟机。
- 支持PF混杂模式、单播列表过滤、组播列表过滤、全组播模式。
- 支持VF单播列表过滤、组播列表过滤、全组播模式。
- 支持VF QinQ模式。
- 支持VF link状态配置为Auto、Enable、Disable。
- 支持VF QoS配置。
- 支持VF MAC地址管理。
- 支持VF spoofchk检查。
- 支持VEB，即Function间支持内部交换。
- 支持RoCEv2卸载。
- 支持UEFI ( Unified Extensible Firmware Interface ) 下PXE ( Preboot eXecution Environment )，在Legacy模式下不支持PXE。支持配置VLAN，支持安全启动，支持BIOS ( Basic Input and Output System ) 下端口界面配置。
- 支持MCTP ( Management Component Transport Protocol )。
- 支持带外方式芯片日志获取。
- 支持网卡管理CLI工具。
- 支持带内一键式日志收集。
- 支持环回测试。

- 支持端口点灯。
- 支持以太端口FEC ( Forward Error Correction ) 模式配置。
- SC371/SC381支持OCP扫描链 ( scan chain ) 。
- 支持以太口自协商。

#### □ 说明

RSS特性：芯片最大支持96个RSS模板，超出部分只能单队列。

## 1.4 技术规格

IN200技术规格如表1-8所示。

表 1-8 技术规格

组件	规格
形态	Low Profile标卡，支持全高和半高两种拉手条
PCIe接口	<ul style="list-style-type: none"><li>● SP570、SP580、SP572和SC371：PCIe x16接口，兼容x8/x4/x2/x1；PCIe Gen3.0，兼容2.0/1.0</li><li>● SP582、SP586和SC381：PCIe x8接口，兼容x4/x2/x1；PCIe Gen3.0，兼容2.0/1.0</li></ul> <p><b>说明</b> 1822网卡的PCIe带宽需要查看PCI bridge。 例如：如下实例需要查看01:00.0来获取带宽信息。 [root@ky6ke4 ~]# lspci  grep 1822 01:00.0 PCI bridge: Huawei Technologies Co., Ltd. Hi1822 Family Virtual Bridge (rev 45) 02:00.0 PCI bridge: Huawei Technologies Co., Ltd. Hi1822 Family Virtual Bridge (rev 45) 02:01.0 PCI bridge: Huawei Technologies Co., Ltd. Hi1822 Family Virtual Bridge (rev 45) 03:00.0 Ethernet controller: Huawei Technologies Co., Ltd. Hi1822 Family (2*100GE) (rev 45) 04:00.0 Ethernet controller: Huawei Technologies Co., Ltd. Hi1822 Family (2*100GE) (rev 45)</p>
网卡芯片	华为海思Hi1822网卡芯片
网络接口	<ul style="list-style-type: none"><li>● SP570和SP580提供4个以太网业务接口（SFP28，25GE/10GE）</li><li>● SP582、SP586和SC381提供2个以太网业务接口（SFP28，25GE/10GE）</li><li>● SP572和SC371提供2个以太网业务接口（QSFP28，100GE/40GE）</li></ul> <p><b>说明</b> SP572/SC371单端口使用时，最大带宽100Gbps。双端口同时使用时，每个端口最大带宽50Gbps。</p>
PXE	<ul style="list-style-type: none"><li>● UEFI模式下支持PXE，Legacy模式下不支持。</li><li>● 仅SP586/SC381/SC371支持安全启动。</li></ul>

组件	规格
平均无故障时间 ( MTBF )	174324小时
平均修复时间 ( MTTR )	180秒

# 2 兼容性

网卡支持的服务器和操作系统请参见[计算产品兼容性查询助手](#)或咨询华为当地销售代表。

# 3 安装与维护

[3.1 获取软件包](#)

[3.2 管理驱动](#)

[3.3 管理固件](#)

## 3.1 获取软件包

### 3.1.1 获取软件包（SPC501 及以下版本）

产品软件包的获取方式，根据用户组的不同而有所差异。

- 对于企业网用户：通过企业网站获取软件包，具体操作请参见[下载软件包](#)。
- 对于运营商用户：联系华为技术有限公司驻当地办事处的技术支持人员。

#### 下载软件包

**步骤1** 登录[IN500 solution产品页面](#)。

**步骤2** 选择“软件”页签。

**步骤3** 打开目标版本。

**步骤4** 下载IN200对应的软件压缩包。

#### 说明

IN200/IN300/IN500软件压缩包都命名为IN500\_solution\_5.1.0.SPCXXX.zip

**步骤5** 解压软件压缩包。需要下载的软件包如[表3-1](#)所示。

**表 3-1** 所需软件包

软件包类型	操作系统类型	路径	格式	安装方法
NIC驱动包	Linux	driver\linux\nic\\操作系统名称\	*.rpm或*.deb	<a href="#">3.2.1.3 安装驱动</a>

软件包类型	操作系统类型	路径	格式	安装方法
	VMware	driver\vmware \nic\操作系统名称 <b>说明</b> <ul style="list-style-type: none"><li>驱动包包含 hinicadm 工具包，驱动安装完成后， hinicadm 工具即自动安装完成。</li><li>驱动包包含一键式日志收集脚本，驱动安装完成后，一键式信息收集脚本即自动安装在 “/opt/huawei/” 路径下。</li><li>一键式日志收集请参见 <a href="#">6.6.1.1 安装日志收集脚本</a>。</li></ul>	*.vib	
RoCE 驱动包	Linux	driver\linux\roce \操作系统名称\	*.rpm	<a href="#">3.2.2.2 安装驱动</a>
固件升级包	Linux/VMware ESXi	<ul style="list-style-type: none"><li>firmware\update_bin\cfg_data_nic_prd_1h_4x25G\</li><li>firmware\update_bin\cfg_data_nic_prd_1h_2x100G\</li><li>firmware\update_bin\cfg_data_nic_prd_1h_2x25G\</li></ul>	*.bin	<a href="#">3.3.2 固件升级</a>
固件日志离线解析字典文件	Linux/VMware ESXi	firmware\dictionary	*.index	-
hinicadm 工具包	Linux	tools\linux\nic\	*.rpm	<a href="#">4.1 安装 hinicadm</a>
	Linux_arm	tools\linux_arm\nic\	*.rpm 或 *.deb	

软件包类型	操作系统类型	路径	格式	安装方法
一键式日志 收集脚本	Linux	tools\linux\nic \collect_scripts	*.sh	<a href="#">6.6.1.1 安装日志收集脚本</a>
	Linux_arm	tools\linux_arm \nic \collect_scripts	*.sh	

----结束

## 注意事项

- 建议使用[华为企业业务网站](#)上发布的最新配套版本的驱动、固件和管理工具。
- 确保安装或升级所使用的固件版本不低于驱动版本，如驱动版本为1.8.2.7，则需要使用1.8.2.7版本或者更高版本的固件。
- 如需进行驱动升级，则需要同时升级配套的固件和管理工具。

## ( 可选 ) 获取 SUSE 系统证书及公钥

SUSE系统对应的驱动包已经过认证，安装此驱动之前，服务器需要先安装对应的证书及公钥。

- [下载SUSE PLDP UEFI证书](#)
- [下载SUSE安装包gpg公钥](#)

## 3.1.2 获取软件包 ( SPC602 及以上版本 )

在安装驱动之前，需要下载驱动配套表、驱动软件。不同类型服务器上网卡驱动包获取路径有所不同。

- 对于智能服务器和鲲鹏服务器，驱动包可在“Computing Component iDriver”下获取。具体获取方法请参见[3.1.2.1 在Computing Component iDriver下下载软件包](#)。
- 对于昇腾计算服务器，驱动包可在各昇腾计算服务器产品路径下获取。具体获取方法请参见[3.1.2.2 在产品路径下下载软件包](#)。

### 3.1.2.1 在 Computing Component iDriver 下下载软件包

在Computing Component iDriver下下载软件包，具体操作如下：

- [下载NIC驱动包和hinicadm工具包](#)
- [下载RoCE驱动包](#)
- [下载固件升级包](#)
- [下载一键式日志收集脚本](#)

#### 下载 NIC 驱动包和 hinicadm 工具包

**步骤1** 单击以下链接，登录Computing Component iDriver页面。

- 企业网用户：[Computing Component iDriver](#)

- 运营商用户：**Computing Component iDriver**

**步骤2** 单击目标“版本或补丁号”。

进入软件列表。

**步骤3** 获取驱动软件包名。

- 根据服务器架构下载ARM或X86的《驱动配套表》。
  - 企业网用户：在页面底部的“版本资料”中下载。
  - 运营商网站用户：在页面底部的“版本文档”中下载。
- 打开《驱动配套表》，选择对应的操作系统的sheet页。
- 通过“Card Name”筛选关键字。  
即可获取不同版本驱动在Computing Component iDriver下压缩包名称以及配套的固件版本，

#### 说明

《驱动配套表》表头各关键字如下所示：

- System Version：驱动适配的操作系统版本。
- External Driver Version：在Computing Component iDriver下驱动文件压缩包的名称。
- Driver File：驱动文件所在镜像名称，解压“External Driver Version”即可获取。
- Onboard ISO Driver contain Files：驱动文件安装包或压缩包，解压或挂载“Driver File”所示镜像文件即可获取。
- Card Name：卡的型号。
- Driver Version：驱动版本。
- FW Version：配套的固件版本。

《驱动配套表》记录着操作系统下所有可安装驱动的部件及对应的驱动信息，如果找不到某个部件的驱动信息，说明在此操作系统下不支持该部件安装。

**步骤4** 获取驱动软件包。

- 在“软件分类”中勾选操作系统类型，如图3-1所示。

**图 3-1 驱动文件**

版本及补丁软件		下载该文件即表示您同意 <a href="#">华为企业软件许可协议</a> 的条款和条件									
软件分类	操作	全部	BC-Linux	CentOS	EulerOS	FW	Kylin	Linx	Other	SLES	UOS
		<input type="checkbox"/> Ubuntu	<input checked="" type="checkbox"/> openEuler								
软件名称	操作	文件大小	发布时间	下载软件次数	下载软件	人工验证签名	自动验证签名				
<input type="checkbox"/> ComputingComponentiDriver-openEuler20.03-Driver-ARM_2.0.0.zip	<a href="#">openEuler</a>	244.35MB	2023-04-28	27	<a href="#">+</a>	<a href="#">pgp</a>	<a href="#">cms</a>				
<input type="checkbox"/> ComputingComponentiDriver-openEuler20.03SP1-Driver-ARM_2.0.0.zip	<a href="#">openEuler</a>	240.07MB	2023-04-28	26	<a href="#">+</a>	<a href="#">pgp</a>	<a href="#">cms</a>				

- 根据《驱动配套表》查到的“External Driver Version”，下载所需操作系统版本的软件压缩包和签名文件。
- 解压获取的压缩包即可得到“External Driver Version”所示的iso文件。
- 解压或挂载iso文件进一步获取驱动软件包。

----结束

### □ 说明

VMware系统下的NIC驱动包包含hinicadm工具包，驱动安装完成后，hinicadm工具即自动安装完成。

## 下载 RoCE 驱动包

联系华为技术有限公司驻当地办事处的技术支持人员。

## 下载固件升级包

**步骤1** 点击以下链接，登录Computing Component iDriver页面。

- 企业网用户：[Computing Component iDriver](#)
- 运营商用户：[Computing Component iDriver](#)

**步骤2** 单击目标“版本或补丁号”。

进入软件列表。

**步骤3** 根据服务器架构在“软件分类”中勾选“FW”、“FW-ARM”或“FW-X86”。

IN200的固件压缩包一般命名为“NIC-*CardName*-1822-FW-x.x.x.x-arch.zip”。

例如：“NIC-SP580-Hi1822-FW-3.9.0.8-X86.zip”或“NIC-SP580-Hi1822-FW-3.9.0.8-ARM.zip”。

### □ 说明

当在指定版本iDriver下无法找到固件包时，可通过搜索关键字（如卡型号或芯片名）查找对应固件包。

**步骤4** 下载对应的固件压缩包及数字签名文件到客户端（本地PC）并解压获取固件文件。

----结束

## 下载一键式日志收集脚本

下载一键式日志收集脚本请参见[3.1.2.1 在Computing Component iDriver下下载软件包](#)。

### □ 说明

- VMware系统下的NIC驱动包驱动包包含一键式日志收集脚本，驱动安装完成后，一键式信息收集脚本即自动安装在“/opt/huawei/”路径下。
- 一键式日志收集请参见[6.6.1 驱动和固件日志收集](#)。

## 注意事项

- 建议使用[华为企业业务网站](#)上发布的最新配套版本的驱动、固件和管理工具。
- 确保安装或升级所使用的固件版本不低于驱动版本，如驱动版本为1.8.2.7，则需要使用1.8.2.7版本或者更高版本的固件。
- 如需进行驱动升级，则需要同时升级配套的固件和管理工具。

## ( 可选 ) 获取 SUSE 系统证书及公钥

SUSE系统对应的驱动包已经过认证，安装此驱动之前，服务器需要先安装对应的证书及公钥。

- [下载SUSE PLDP UEFI证书](#)
- [下载SUSE安装包gpg公钥](#)

### 3.1.2.2 在产品路径下下载软件包

产品软件包的获取方式，根据用户组的不同而有所差异。

- 对于企业网用户：通过企业网站获取软件包，具体操作请参见本章节。

软件包	下载方式
NIC驱动包和hinicadm工具包	<a href="#">下载NIC驱动包和hinicadm工具包</a>
RoCE驱动包	<a href="#">下载RoCE驱动包</a>
固件升级包	<a href="#">下载固件升级包</a>
一键式日志收集脚本	<a href="#">下载一键式日志收集脚本</a>

- 对于运营商用户：联系华为技术有限公司驻当地办事处的技术支持人员。

### 下载 NIC 驱动包和 hinicadm 工具包

**步骤1** 登录[昇腾计算服务器产品页面](#)。

**步骤2** 选择服务器型号。

**步骤3** 单击“软件”页签。

进入服务器软件所在页面。

**步骤4** 在“版本资料”列表中下载《版本配套表》。

《版本配套表》中“软件包名称”可以查看到不同操作系统对应的软件包名称。

#### 说明

《版本配套表》记录着所有OS下可安装组件及对应的版本信息，如果找不到组件的信息，说明在此OS下不支持该组件安装。

**步骤5** 在软件列表下载软件压缩包<Server Name>-<OS>-Driver-<version>.zip到客户端（本地PC）并解压得到软件iso文件onboard\_driver\_<OS>.iso。

**步骤6** 解压软件iso文件“onboard\_driver\_<OS>.iso”获取NIC驱动包和hinicadm工具包：

- 驱动软件包“NIC-IN200-<OS>-hinic-<version>-<OS Architecture>.rpm”或“NIC-IN200-<OS>-hinic-<version>-<OS Architecture>.deb”
- hinicadm工具包“NIC-IN200-<OS>-hinicadm-<version>-<OS Architecture>.rpm”或“NIC-IN200-<OS>-hinicadm-<version>-<OS Architecture>.deb”

----结束

## 下载 RoCE 驱动包

联系华为技术有限公司驻当地办事处的技术支持人员。

## 下载固件升级包

**步骤1** 登录[昇腾计算服务器产品页面](#)。

**步骤2** 选择服务器型号。

**步骤3** 单击“软件”页签。

进入服务器软件所在页面。

**步骤4** 选择目标版本。

**步骤5** 在“版本资料”列表中下载《版本配套表》。

《版本配套表》中“软件包名称”可以查看到不同操作系统对应的软件包名称。

### □ 说明

《版本配套表》记录着所有OS下可安装组件及对应的版本信息，如果找不到组件的信息，说明在此OS下不支持该组件安装。

**步骤6** 在软件列表中下载软件压缩包<Server Name>-NIC-<Card Name>-1822-FW-<OS Architecture>.zip到客户端（本地PC）并解压获得固件升级包“XXX.bin”。

----结束

## 下载一键式日志收集脚本

下载一键式日志收集脚本请参见[下载软件包](#)。

### □ 说明

- VMware系统下的NIC驱动包驱动包包含一键式日志收集脚本，驱动安装完成后，一键式信息收集脚本即自动安装在“/opt/huawei/”路径下。
- 一键式日志收集请参见[6.6.1 驱动和固件日志收集](#)。

## 注意事项

- 建议使用[华为企业业务网站](#)上发布的最新配套版本的驱动、固件和管理工具。
- 确保安装或升级所使用的固件版本不低于驱动版本，如驱动版本为1.8.2.7，则需要使用1.8.2.7版本或者更高版本的固件。
- 如需进行驱动升级，则需要同时升级配套的固件和管理工具。

## （可选）获取 SUSE 系统证书及公钥

SUSE系统对应的驱动包已经过认证，安装此驱动之前，服务器需要先安装对应的证书及公钥。

- [下载SUSE PLDP UEFI证书](#)
- [下载SUSE安装包gpg公钥](#)

### 3.1.3 软件完整性验证

为了防止软件包在传递过程或存储期间被恶意篡改，华为提供软件包PGP（后缀名.asc）和CMS（后缀名.p7s）两种格式的数字签名，分别用于工程师手工验证和系统自动验证。下载本软件即表示您同意[华为企业软件许可协议](#)的条款和条件。

1. 导入软件包前，手动验证PGP格式签名，确保软件包未被篡改。
2. 导入软件包时，同步提供软件包的CMS格式签名，进行数字签名校验，确保软件包未被篡改。

#### 说明

如果校验失败，请重新下载软件包和签名。如仍校验失败，请联系华为技术支持工程师解决。

运营商客户请访问：[《OpenPGP签名验证指南》](#)

企业客户请访问：[《OpenPGP签名验证指南》](#)

## 3.2 管理驱动

### 3.2.1 NIC 驱动维护

#### 3.2.1.1 使用一键脚本安装升级驱动

##### 前提条件

已下载网卡的驱动包。

获取方法可参见[3.1 获取软件包](#)。

##### 影响

安装驱动所需时间在30s左右，安装过程不可中断。在此期间OS不能重启，否则可能造成OS异常、无法正常启动等情况。

##### 一键安装或升级驱动

#### 说明

当服务器下存在多个同型号网卡时，使用一键安装驱动脚本将为所有网卡安装/升级驱动。

**步骤1** 登录服务器OS。

**步骤2** 挂载升级文件。

#### 说明

iBMC远程虚拟控制台包含Java远程虚拟控制台和HTML5集成远程控制台，其中HTML5集成远程控制台不支持“本地文件夹”功能，本章节以Java远程虚拟控制台为例进行说明，更多信息可参考[iBMC用户指南](#)。

1. 在虚拟控制台的工具栏中单击，打开虚拟光驱下属菜单，如图3-2所示。

图 3-2 虚拟光驱



2. 选择“镜像文件”并单击“浏览”，打开文件选择窗口。
3. 选择驱动包解压后得到的“onboard\_driver\_xxx.iso”文件。
4. 单击“连接”。

当工具栏的“弹出”为可操作状态时，即表示已经成功挂载，如图3-3所示。

图 3-3 挂载成功



### 步骤3 拷贝文件到服务器，并为其赋予读写权限。

1. 执行`mount /dev/sr0 /mnt`命令，将本地文件夹挂载到“/mnt”路径下。

#### 说明

可在输入`mount /dev/sr`后，按Tab键，自动联想起sr后面的值。

2. 执行`mkdir /root/driver`命令，在root目录下新建“driver”文件夹。
3. 执行`cp -vr /mnt/* /root/driver/`命令将所挂载的文件拷贝到“driver”目录下。
4. 执行`cd /root/driver/`命令进入“driver”目录。
5. 执行`chmod +x install.sh`为脚本添加可执行权限。

### 步骤4 执行`./install.sh`执行自动安装脚本。

### 步骤5 进入驱动安装界面。包含如下选项：

- 1：自动安装目录下所有驱动和`hinicadm3`命令行工具。
- 2：查看当前网卡信息和驱动版本。
- H：帮助信息。
- Q：退出当前菜单。

输入“1”，并按“Enter”，执行一键安装所有驱动。

### 步骤6 安装完成后，按“Enter”，返回选项菜单。

- 输入“2”，查看当前网卡信息和驱动版本。
- 输入“q”，并按“Enter”，退出驱动安装界面。

### 步骤7 生效驱动。

安装完毕后，可对OS执行`reboot`命令，使新的驱动生效。

----结束

### 3.2.1.2 准备工作（SUSE）

由于IN200已经过SUSE认证，在对SUSE系统对应的驱动做安装、升级操作之前，需要先执行SUSE PLDP（Partner Linux Driver Program）UEFI证书导入（Secure Boot模式下必选）、安装包gpg公钥导入的操作。

#### （可选）导入 SUSE PLDP UEFI 证书

仅当x86服务器在BIOS UEFI安全模式下启动时，为配合已认证的IN200的驱动操作，需要在BIOS中导入SUSE PLDP UEFI证书。

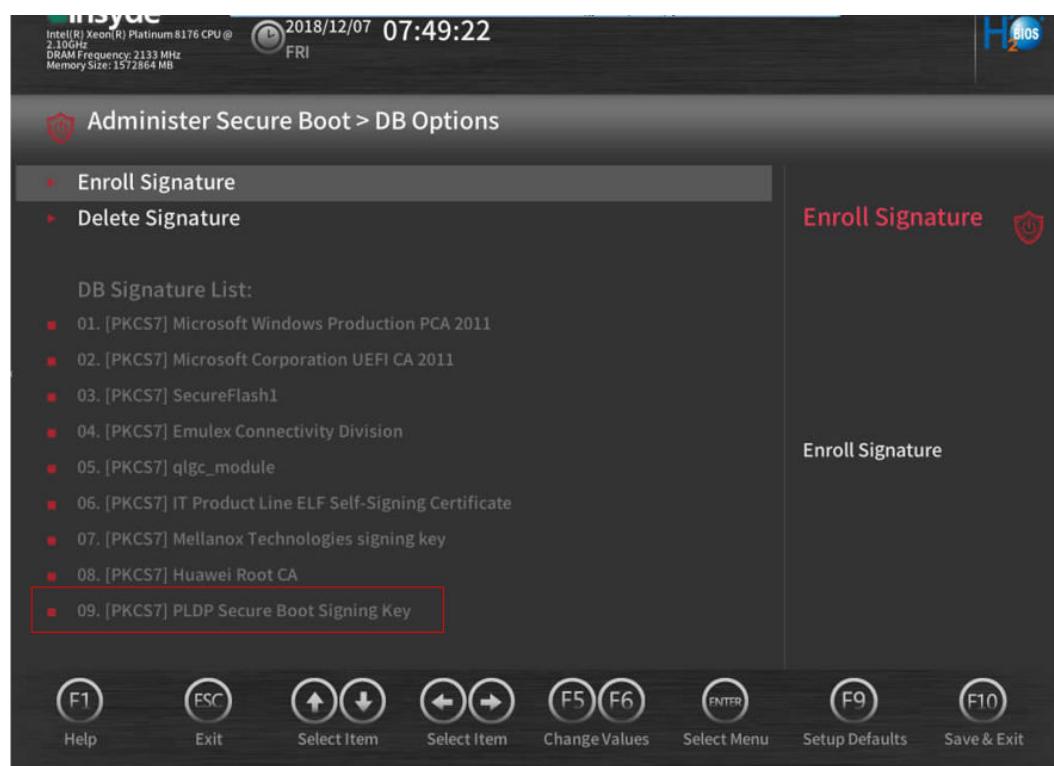
**步骤1** 获取SUSE PLDP UEFI证书。

证书获取地址请参见[3.1 获取软件包](#)。

**步骤2** 将获取到的SUSE PLDP UEFI证书导入到BIOS中。

在BIOS界面选择“Administer Secure Boot > DB Options > Enroll Signature”，导入SUSE PLDP UEFI证书，证书导入后在DB Signature List列表中会出现如图3-4红框所示的PLDP Secure Boot Signing Key。

图 3-4 导入证书



详细操作方法可参见服务器BIOS参数参考。

----结束

## 导入 SUSE 安装包 gpg 公钥

### 须知

SUSE安装包gpg公钥只需导入一次即可。

**步骤1** 获取SUSE安装包gpg公钥。

证书获取地址请参见[3.1 获取软件包](#)。

**步骤2** 将获取到的公钥文件（例如“gpg-pubkey-c2bea7e6-4c2de264.asc”）传输到服务器OS的任意目录下。

**步骤3** 执行**rpm --import**命令将公钥导入OS。

```
rpm --import gpg-pubkey-c2bea7e6-4c2de264.asc
```

----结束

### 3.2.1.3 安装驱动

#### 前提条件

- 已下载IN200网卡的驱动包。  
NIC驱动包包含在IN200的整体驱动包中，获取方法可参见[3.1 获取软件包](#)。
- 若操作的是SUSE系统对应的驱动，需首先完成准备工作。
- 驱动包已上传到服务器OS。

#### 影响

安装驱动所需时间在30s左右，安装过程不可中断。在此期间OS不能重启，否则可能造成OS异常、无法正常启动等情况。

### Linux 系统下安装驱动

**步骤1** 登录服务器OS。

**步骤2** 进入驱动包所在目录。

**步骤3**（可选）PF直通虚拟机场景，虚拟机中需要单独配置参数。

执行echo "options hinic l2nic\_interrupt\_switch=N" >> /etc/modprobe.d/hinic.conf命令关闭中断。

**步骤4** 执行命令安装NIC驱动。

- 包名为\*.rpm时：

执行**rpm -ivh <驱动软件包名称>**安装驱动。

以RHEL系统为例：

```
[root@localhost]# rpm -ivh kmod-hinic-1.8.2.8_3.10.0_957-1.el7.x86_64.rpm
Preparing... ################################ [100%]
Updating / installing...
1:kmod-hinic-1.8.2.8_3.10.0_957-1.e######################################## [100%]
```

- 包名为\*.deb时：  
执行**dpkg -i <驱动软件包名称>**安装驱动。  
以Ubuntu系统为例：

```
root@ubuntu1804:/home/ubuntu # dpkg -i hinic-1.6.1.1-4.15.0_20_generic.ubuntu.arm64.deb
Selecting previously unselected package hinic.
(Reading database ... 69638 files and directories currently installed.)
Preparing to unpack hinic-1.6.1.1-4.15.0_20_generic.ubuntu.arm64.deb ...
Unpacking hinic (1.6.1.1) ...
Setting up hinic (1.6.1.1) ...
Installing... Please wait for a moment.
Install hinic driver package successfully.
```

## 步骤5 使驱动生效。

驱动生效方式有两种，请按照实际需要进行选择。

- 方法一：  
重启OS。
- 方法二：
  - a. 执行**lsmod |grep hinic**查看hinic是否已加载。
    - 是，执行**步骤5.b**。
    - 否，执行**步骤5.c**。
  - b. 执行**rmmmod hinic**卸载当前正在运行的驱动。
  - c. 执行**modprobe hinic**加载已安装的驱动。

----结束

## VMware ESXi 系统下安装驱动 (\*.vib )

步骤1 登录服务器OS。

步骤2 进入驱动包所在目录，例如“/tmp”。

步骤3 安装驱动包。

执行**esxcli software vib install -v hinic-<version>-<kernel\_version>. <arch>.vib**安装驱动包。

### 说明

-v后面的要填完整路径。

例如：

```
[root@localhost:-] esxcli software vib install -v /hinic-1.6.2.2-1OEM.650.0.0.4598673.x86_64.vib
Installation Result
Message: The update completed successfully, but the system needs to be rebooted for the changes to be effective.
Reboot Required: true
VIBs Installed: Huawei_bootbank_hinic_1.6.2.2-1OEM.650.0.0.4598673
VIBs Removed:
VIBs Skipped:
```

步骤4 在VMware ESXi下执行**reboot**命令，使驱动生效。

----结束

## VMware ESXi 系统下安装驱动 (\*.zip)

步骤1 登录服务器OS。

步骤2 进入驱动包所在目录，例如 “/tmp”。

步骤3 安装驱动包。

执行`esxcli software vib install -d /tmp/*.zip`安装驱动包。

### □ 说明

-d后面的要填完整路径。

例如：

```
[root@localhost:/tmp] esxcli software vib install -d /tmp/VMW-ESX-6.5.0-hinic-3.2.0.17-offline_bundle-17294290.zip
Installation Result
    Message: The update completed successfully, but the system needs to be rebooted for the changes to be effective.
    Reboot Required: true
    VIBs Installed: Huawei_bootbank_hinic_3.2.0.17-1OEM.650.0.0.4598673
    VIBs Removed:
    VIBs Skipped:
```

步骤4 在VMware ESXi下执行`reboot`命令，使驱动生效。

----结束

## 后续操作

驱动安装成功后，您可以通过下列操作确定驱动生效情况。

- 在Linux下执行`lsmod | grep hinic`，或在VMware ESXi下执行`vmkload_mod -l | grep hinic`命令，通过回显信息查看驱动是否已生效。
  - 若无回显，说明驱动尚未生效，请重新操作。
  - 若有IN200的驱动相关回显，说明驱动已生效。
- 通过hinicadm工具的`hinicadm version -i <devicename>`命令查询生效驱动的版本信息，其中`<devicename>`为IN200在操作系统中的设备名称，例如`hinic0`。

### 3.2.1.4 升级驱动

#### 前提条件

- 服务器OS当前已存在IN200的驱动。
- 客户端已下载IN200网卡的驱动包。  
NIC驱动包包含在IN200的整体驱动包中，获取方法可参见[3.1 获取软件包](#)。
- 若操作的是SUSE系统对应的驱动，需首先完成[准备工作](#)。
- 驱动包已上传到服务器OS。

#### 影响

升级过程不可中断。在此期间OS不能重启，否则可能造成OS异常、无法正常启动等情况。

## Linux 系统下升级驱动

**步骤1** 登录服务器OS。

**步骤2** 进入驱动包所在目录。

**步骤3** 升级驱动包。

- 包名为\*.rpm时：

执行**rpm -Uvh <驱动软件包名称>**升级驱动。

例如：

```
[root@localhost]# rpm -Uvh kmod-hinic-2.3.0.0_3.10.0_957-1.el7.x86_64.rpm
Preparing... ################################ [100%]
Updating / installing...
1:kmod-hinic-2.3.0.0_3.10.0_957-1.e######################################## [ 50%]
Cleaning up / removing...
2:kmod-hinic-1.8.2.8_3.10.0_957-1.e######################################## [100%]
```

- 包名为\*.deb时：

执行**dpkg -i <驱动软件包名称>**升级驱动。

例如：

```
root@ubuntu1804:/home/ubuntu # dpkg -i hinic-1.6.1.2-4.15.0_20_generic.ubuntu.arm64.deb
(Reading database ... 69641 files and directories currently installed.)
Preparing to unpack hinic-1.6.1.2-4.15.0_20_generic.ubuntu.arm64.deb ...
Unpacking hinic (1.6.1.2) over (1.6.1.1) ...
Uninstalling... Please wait for a moment.
Uninstall hinic driver package successfully.
Setting up hinic (1.6.1.2) ...
Installing... Please wait for a moment.
Install hinic driver package successfully.
```

**步骤4** 使新的驱动生效。

驱动生效方式有两种，请按照实际需要进行选择。

- 方法一：

在OS下执行**reboot**命令。

- 方法二：

a. 确认使用当前IN200的应用程序已停止。

b. 在OS下执行**rmmmod hinic**命令卸载当前运行的旧驱动。

c. 在OS下执行**modprobe hinic**命令加载已安装的新驱动。

----结束

## VMware ESXi 系统下升级驱动

升级驱动直接按照[3.2.1.3 安装驱动](#)操作进行即可，系统会根据版本号的更新自动替换旧的vib驱动包。

## 后续操作

在驱动升级完成后，可以通过hinicadm工具的**hinicadm version -i <devicename>**命令查询生效驱动的版本信息，其中<devicename>为IN200在操作系统中的设备名称，例如hinic0。

### 3.2.1.5 卸载驱动

#### 前提条件

- OS中已存在IN200的驱动。
- 若操作的是SUSE系统对应的驱动，需首先完成3.2.1.2 准备工作（SUSE）。

#### 影响

卸载驱动过程不可中断，否则可能造成OS异常、无法正常启动等情况。

### Linux 系统下卸载驱动

**步骤1** 登录服务器OS。

**步骤2** 进入驱动的安装目录。

**步骤3** 卸载驱动。

- 包名为\*.rpm时，执行**rpm -e \***命令卸载驱动。

以RHEL为例：

```
rpm -e kmod-hinic-1.2.3.0_3.10.0_514-1.el7.x86_64
```

#### □ 说明

驱动软件包名称可以通过**rpm -qa | grep hinic**查询。

- 包名为\*.deb时，执行**dpkg -r hinic**命令卸载驱动。

以Ubuntu操作系统为例：

```
root@ubuntu1804:~# dpkg -r hinic
(Reading database ... 65985 files and directories currently installed.)
Removing hinic (1.8.2.3) ...
Uninstalling... Please wait for a moment.
Uninstall hinic driver package successfully.
```

**步骤4** 使卸载驱动的操作生效。

卸载驱动生效方式有两种，请按照实际需要进行选择。

- 方法一：

在OS下执行**reboot**命令。

- 方法二：

a. 确认使用当前IN200的应用程序已停止。

b. 在OS下执行**rmmmod hinic**命令。

----结束

### VMware ESXi 系统下卸载驱动

**步骤1** 登录服务器OS。

**步骤2** 进入驱动的安装目录。

**步骤3** 卸载驱动。

执行**esxcli software vib remove -n hinic**卸载hinic驱动，例如：

```
[root@localhost:-] esxcli software vib remove -n hinic
Removal Result
  Message: The update completed successfully, but the system needs to be rebooted for the changes to be
effective
  Reboot Required: true
  VIBs Installed:
  VIBs Removed:Huawei_bootbank_hinic_1.6.2.1-1OEM.650.0.0.4598673
  VIBs Skipped:
```

步骤4 执行reboot命令，使驱动完全卸载。

----结束

## 3.2.2 RoCE 驱动维护

### 3.2.2.1 RoCE 支持的操作系统

当您需要在IN200上运行RoCE（RDMA over Converged Ethernet）协议相关业务时，需要先为其安装对应OFED和RoCE驱动。

OFED软件包分为HW OFED和OFA OFED两种。HW OFED包含在RoCE驱动包中，RoCE支持的操作系统如表3-2所示。

表 3-2 RoCE 支持的操作系统

操作系统	x86	ARM
BC-Linux	-	7.6
CentOS	7.4/7.5/7.6/8.0/8.0_ini/8. 1/8.2/8.2_ini  说明 CentOS 8.X_ini使用OS自带 OFED，无需执行安装OFED 的步骤。	7.5/7.6/8.0/8.1/8.2
EulerOS	V200R005C00SPC310B05 6/ V200R007C00SPC500B00 5	V200R008C00SPC106B350/ V200R008C00SPC306B830/ V200R009C00SPC100B150
NeoKylin（中标麒麟）	-	7.6
OpenEuler	-	20.3
RHEL	7.5/7.6/8.0	-
SUSE	12.2/12.4	12.4/15.1
Ubuntu	16.04.4/18.04.1	18.04.1/18.04.2/18.04.3
UOS	-	20SP1

### 3.2.2.2 安装驱动

#### 前提条件

- 已下载IN200网卡的RoCE驱动包。  
RoCE驱动包包含在IN200的整体驱动包中，获取方法请参见[3.1 获取软件包](#)。
- 服务器OS上已安装了IN200对应的NIC驱动包。  
安装方法请参见[NIC驱动维护](#)。
- 驱动包已上传到服务器OS。

#### 影响

- 在服务器已安装非OFA OFED 4.8-2版本软件包（例如第三方OFED版本或其他开源OFED版本的软件包）的场景下，若使用IN200 RoCE功能，需要先卸载已安装的OFED软件包，再安装OFA OFED 4.8-2及IN200 RoCE驱动。此场景与IN200 NIC功能不冲突。
- 安装驱动所需时间在30s左右，安装过程不可中断。在此期间OS不能重启，否则可能造成OS异常、无法正常启动等情况。

### HW OFED 支持的操作系统下安装 RoCE 驱动

**步骤1** 登录服务器OS。

**步骤2** 执行`rpm -qa | grep rdma`和`rpm -qa | grep libib`或`dpkg -l | grep rdma`和`dpkg -l | grep libib`查看系统中是否已经安装OFED。

- 是，执行**步骤3**。
- 否，执行**步骤4**。

**步骤3** 执行`rpm -e X`或`dpkg -r X`卸载原有OFED，`X`为**步骤2**回显的所有OFED包名称。

**步骤4** 在RoCE驱动包所在目录执行如下命令解压OFED压缩包。

`tar -xvf OFED-*.tar.gz`

**步骤5** 执行如下命令进入OFED包目录。

`cd OFED-*`

**步骤6** 执行如下命令安装OFED包。

`bash ofed_install.sh`

回显如下信息：

```
2019-07-10 09:30:43 [INSTALL] install OFED software start.
2019-07-10 09:30:43 [INSTALL] check RPMS/compat-rdma-devel*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/compat-rdma*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/rdma-core-devel*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/rdma-core*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/libverbs*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/libverbs-utils*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/libbumad*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/librdmacm*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/librdmacm-utils*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/perftest-debuginfo*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/perftest*rpm ok
This program will install the OFED package on your machine.
```

```
Note that all other Mellanox, OEM, OFED, RDMA or Distribution IB packages will be removed.  
Those packages are removed due to conflicts with OFED, do not reinstall them.  
Do you want to continue?[y/N]:y
```

```
.....  
Preparing... ###### [100%]  
Updating / installing...  
 1:perf-test-debuginfo-4.4-0.5.g1ceab#### [ 50%]  
 2:perf-test-4.4-0.5.g1ceab48 #### [100%]  
2019-07-10 09:33:26 [INSTALL] install_rpm perf-test ok  
2019-07-10 09:33:26 [INSTALL] install driver -----pass  
2019-07-10 09:33:26 [INSTALL] install OFED software ok.  
To load the new driver, run:  
/etc/init.d/openibd restart
```

### 步骤7 在RoCE驱动包所在目录安装RoCE驱动。

- 包名为\*.rpm时：

执行**rpm -ivh <驱动软件包名称>**安装RoCE驱动。

以CentOS 7.4系统为例：

```
[root@localhost driver]# rpm -ivh hiroce-2.3.1.0_3.10.0_693.el7.x86_64-1.el7.centos.x86_64.rpm  
Preparing... ###### [100%]  
Updating / installing...  
1:hiroce-2.3.1.0_3.10.0_693.el7.x86#### [100%]
```

- 包名为\*.deb时：

执行**dpkg -i <驱动软件包名称>**安装RoCE驱动。

以Ubuntu 18.04.1系统为例：

```
root@ubuntu18041:/home/ubuntu # dpkg -i hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb  
Selecting previously unselected package hiroce.  
(Reading database ... 80445 files and directories currently installed.)  
Preparing to unpack hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb ...  
Unpacking hiroce (2.3.0.2) ...  
Setting up hiroce (2.3.0.2) ...  
Installing... Please wait for a moment.  
Install hiroce driver package successfully.
```

### 步骤8 使驱动生效。

在OS下执行如下命令。

**service openibd restart**



说明

该方式为临时生效方式，建议重启操作系统。

----结束

## OFA OFED 支持的操作系统下安装 RoCE 驱动

### 步骤1 登录服务器OS。

### 步骤2 执行**rpm -qa | grep rdma**和**rpm -qa | grep libib**或**dpkg -l | grep rdma**和**dpkg -l | grep libib**查看系统中是否已经安装OFED。

- 是，执行**步骤3**。
- 否，执行**步骤4**。

### 步骤3 执行**rpm -e X**或**dpkg -r X**卸载原有OFED，X为**步骤2**回显的所有OFED包名称。

**步骤4** 在[OFA OFED-4.8-2](#)下载4.8-2版本的OFA OFED压缩包。

**步骤5** 参考OFA OFED压缩包中的README.txt编译出对应rpm/deb包（包括compat-rdma和rdma-core）。

**步骤6** 参见[B 通过虚拟文件夹传输文件](#)将rpm/deb包上传到操作系统。

**步骤7** 执行**rpm -ivh \*.rpm**或**dpkg -i \*.deb**安装OFA OFED 4.8-2。

**步骤8** 在RoCE驱动包所在目录安装RoCE驱动。

- 包名为\*.rpm时：

执行**rpm -ivh <驱动软件包名称>**安装驱动包。

以RHEL 7.3系统为例：

```
[root@localhost driver]# rpm -ivh hiroce-2.3.1.0_3.10.0_514.el7.x86_64-1.el7.x86_64.rpm
Preparing... ################################ [100%]
Updating / installing...
1:hiroce-2.3.1.0_3.10.0_514.el7.x86#### [100%]
```

- 包名为\*.deb时：

执行**dpkg -i <驱动软件包名称>**安装RoCE驱动。

以Ubuntu 18.04.1系统为例：

```
root@ubuntu18041:/home/ubuntu # dpkg -i hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb
Selecting previously unselected package hiroce.
(Reading database ... 80445 files and directories currently installed.)
Preparing to unpack hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb ...
Unpacking hiroce (2.3.0.2) ...
Setting up hiroce (2.3.0.2) ...
Installing... Please wait for a moment.
Install hiroce driver package successfully.
```

**步骤9** 使驱动生效。

在OS下执行如下命令。

**service openibd restart**

#### □ 说明

该方式为临时生效方式，建议重启操作系统。

#### ----结束

## 自带 OFED 支持的操作系统下安装 RoCE 驱动

**步骤1** 登录服务器OS。

**步骤2** 在RoCE驱动包所在目录安装RoCE驱动。

执行**rpm -ivh <驱动软件包名称>**安装驱动包。

以CentOS 8.2系统为例：

```
[root@localhost driver]# rpm -ivh hiroce-3.9.0.5_4.18.0_193.el8.x86_64-1.el8.x86_64.rpm
Verifying... ################################ [100%]
Preparing... ################################ [100%]
Updating / installing...
1:hiroce-3.9.0.5_4.18.0_193.el8.x86#### [100%]
```

**步骤3** 使驱动生效。

执行重启操作系统命令。

**reboot**

----结束

## 后续操作

驱动安装成功后，您可以通过下列操作确定驱动生效情况。

1. 在OS下执行`lsmod | grep hiroce`命令，通过回显信息查看驱动是否已生效。
  - 若有IN200的驱动相关回显，说明驱动已生效。
  - 若无回显，说明驱动尚未生效，请重新操作。
2. 执行`hiroce gids`命令，通过回显信息查看驱动是否已生效。
  - 若有设备回显信息，说明驱动已生效。
  - 若无回显，说明驱动尚未生效，请重新操作。

### 3.2.2.3 升级驱动

#### 前提条件

- 已下载IN200网卡的最新版本RoCE驱动包。  
RoCE驱动包包含在IN200的整体驱动包中，获取方法请参见[3.1 获取软件包](#)。
- 服务器OS上已安装了IN200对应的NIC驱动包。  
安装方法请参见[NIC驱动维护](#)。
- 服务器OS上已安装RoCE驱动。
- 驱动包已上传到服务器OS。

#### 约束

- 待升级驱动包版本必须较已加载驱动包版本高。
- 如待升级驱动包版本较已加载驱动包版本低或相同，使用`rpm -Uvh`命令升级后无法生效。
- 如想退回到低版本驱动包，需要先根据[3.2.2.4 卸载驱动](#)卸载驱动后，再根据[3.2.2.2 安装驱动](#)安装驱动。

#### 影响

升级过程不可中断。在此期间OS不能重启，否则可能造成OS异常、无法正常启动等情况。

### HW OFED 支持的操作系统下升级 RoCE 驱动

**步骤1** 登录服务器OS。

**步骤2** 在RoCE驱动包所在目录执行如下命令解压OFED压缩包。

```
tar -xvf OFED-*.tar.gz
```

**步骤3** 执行如下命令进入OFED包目录。

```
cd OFED-*
```

**步骤4** 执行如下命令安装OFED包。

**bash ofed\_install.sh**

回显如下信息：

```
[root@localhost OFED]# bash ofed_install.sh
2019-07-10 09:30:43
2019-07-10 09:30:43
2019-07-10 09:30:43 [INSTALL] install OFED software start.
2019-07-10 09:30:43 [INSTALL] check RPMS/compat-rdma-devel*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/compat-rdma*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/rdma-core-devel*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/rdma-core*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/libibverbs*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/libibverbs-utils*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/libbumad*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/librdmacm*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/librdmacm-utils*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/perftest-debuginfo*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/perftest*rpm ok
This program will install the OFED package on your machine.
Note that all other Mellanox, OEM, OFED, RDMA or Distribution IB packages will be removed.
Those packages are removed due to conflicts with OFED, do not reinstall them.
Do you want to continue?[y/N]:y
.....
.....
.....
Preparing... ##### [100%]
Updating / installing...
1:perftest-debuginfo-4.4-0.5.g1ceab##### [50%]
2:perftest-4.4-0.5.g1ceab48 ##### [100%]
2019-07-10 09:33:26 [INSTALL] install_rpm perftest ok
2019-07-10 09:33:26 [INSTALL] install driver -----pass
2019-07-10 09:33:26 [INSTALL] install OFED software ok.
To load the new driver, run:
/etc/init.d/openibd restart
```

**步骤5** 在RoCE驱动包所在目录升级驱动包。

- 包名为\*.rpm时：

执行**rpm -Uvh <驱动软件包名称>**升级RoCE驱动。

以RHEL 7.3系统为例：

```
[root@localhost driver]# rpm -Uvh hiroce-2.3.1.0_3.10.0_514.el7.x86_64-1.el7.x86_64.rpm
Preparing... ##### [100%]
Updating / installing...
1:hiroce-2.3.1.0_3.10.0_514.el7.x86##### [100%]
```

- 包名为\*.deb时：

执行**dpkg -i <驱动软件包名称>**升级RoCE驱动。

以Ubuntu 18.04.1系统为例：

```
root@ubuntu18041:/home/ubuntu # dpkg -i hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb
Selecting previously unselected package hiroce.
(Reading database ... 80445 files and directories currently installed.)
Preparing to unpack hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb ...
Unpacking hiroce (2.3.0.2) ...
Setting up hiroce (2.3.0.2) ...
Installing... Please wait for a moment.
Install hiroce driver package successfully.
```

**步骤6** 使新的驱动生效。

在OS下执行如下命令。

**rmmmod hiroce**

**modprobe hiroce****□ 说明**

- 使新驱动生效前，若升级前为RoCE bonding场景，需要修改/etc/modprobe.d/hiroce\_pro.conf文件，修改方式见[6.4.3.1 通过命令行配置RoCE bonding步骤3。](#)
- 该方式为临时生效方式，建议重启操作系统。

**----结束****OFA OFED 支持的操作系统下升级 RoCE 驱动****步骤1** 登录服务器OS。**步骤2** 在[OFA OFED-4.8-2](#)下载4.8-2版本的OFA OFED压缩包。**步骤3** 参考OFA OFED压缩包中的README.txt编译出对应rpm/deb包（包括compat-rdma和rdma-core）。**步骤4** 参见[B 通过虚拟文件夹传输文件](#)将rpm/deb包上传到操作系统。**步骤5** 执行**rpm -ivh \*.rpm**或**dpkg -i \*.deb**安装OFA OFED。**步骤6** 在RoCE驱动包所在目录升级RoCE驱动。

- 包名为\*.rpm时：

执行**rpm -Uvh <驱动软件包名称>**升级RoCE驱动。

以RHEL 7.3系统为例：

```
[root@localhost driver]# rpm -Uvh hiroce-2.3.1.0_3.10.0_514.el7.x86_64-1.el7.x86_64.rpm
Preparing... ################################ [100%]
Updating / installing...
1:hiroce-2.3.1.0_3.10.0_514.el7.x86_64 ################################ [100%]
```

- 包名为\*.deb时：

执行**dpkg -i <驱动软件包名称>**升级RoCE驱动。

以Ubuntu 18.04.1系统为例：

```
root@ubuntu18041:/home/ubuntu # dpkg -i hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb
Selecting previously unselected package hiroce.
(Reading database ... 80445 files and directories currently installed.)
Preparing to unpack hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb ...
Unpacking hiroce (2.3.0.2) ...
Setting up hiroce (2.3.0.2) ...
Installing... Please wait for a moment.
Install hiroce driver package successfully.
```

**步骤7** 使新的驱动生效。

在OS下执行如下命令。

**rmmod hiroce****modprobe hiroce****□ 说明**

- 使新驱动生效前，若升级前为RoCE bonding场景，需要修改/etc/modprobe.d/hiroce\_pro.conf文件，修改方式见[6.4.3.1 通过命令行配置RoCE bonding步骤3。](#)
- 该方式为临时生效方式，建议重启操作系统。

**----结束**

## 自带 OFED 支持的操作系统下升级 RoCE 驱动

**步骤1** 登录服务器OS。

**步骤2** 在RoCE驱动包所在目录升级RoCE驱动。

执行`rpm -Uvh <驱动软件包名称>`升级RoCE驱动。

以CentOS 8.2系统为例：

```
[root@localhost driver]# rpm -Uvh hiroce-3.9.0.5_4.18.0_193.el8.x86_64-1.el8.x86_64.rpm
Verifying... ###### [100%]
Preparing... ###### [100%]
Updating / installing...
1:hiroce-3.9.0.5_4.18.0_193.el8.x86#### [100%]
```

**步骤3** 使新的驱动生效。

在OS下执行如下命令。

`rmmmod hiroce`

`modprobe hiroce`

### 说明

- 使新驱动生效前，升级前为RoCE bonding场景，需要修改`/etc/modprobe.d/hiroce_pro.conf`文件，修改方式见[6.4.3.1 通过命令行配置RoCE bonding](#)步骤3。
- 该方式为临时生效方式，建议重启操作系统。

----结束

### 3.2.2.4 卸载驱动

#### 前提条件

OS中已存在IN200的驱动。

#### 影响

卸载驱动过程不可中断，否则可能造成OS异常、无法正常启动等情况。

#### 操作步骤

##### 须知

卸载前使用RoCE的应用软件需全部关闭，否则驱动会被占用，导致卸载失败。

**步骤1** 登录服务器OS。

**步骤2** 进入驱动的安装目录。

**步骤3** 卸载驱动。

- 对于包名为\*.rpm的驱动，执行`rpm -e hiroce`命令卸载。

### □ 说明

驱动软件包名称可以通过`rpm -qa | grep hiroce`查询。

- 对于包名为\*.deb的驱动，执行`dpkg -r hiroce`命令卸载。

### □ 说明

驱动软件包名称可以通过`dpkg -l | grep hiroce`查询。

#### 步骤4 (可选) 卸载OFED。

- HW OFED支持的操作系统下，进入OFED软件包执行`bash uninstall.sh`或`bash ofed_uninstall.sh`命令卸载驱动OFED。
- OFA OFED支持的操作系统下，执行`rpm -e X`或`dpkg -r X`卸载OFED。
- 自带OFED支持的操作系统下，跳过此步骤。

#### 步骤5 使卸载驱动的操作生效。

卸载驱动生效方式有两种，请按照实际需要进行选择。

- 方法一：  
在OS下执行`reboot`命令。
- 方法二：
  - 确认使用当前IN200的RoCE应用已停止。
  - 在OS下执行`rmmmod hiroce`命令。

----结束

## 3.3 管理固件

### □ 说明

升级固件前，请查询“Secure Boot Status”参数值：

- 若为“Disabled”，请直接升级。
- 若为“Enabled”，请将“Secure Boot Status”参数设置为“Disabled”后再进行网卡升级，网卡升级完成后，再将参数设置为“Enabled”。

具体查询和设置“Secure Boot Status”参数值的方法，请参考《华为服务器 Grantley平台 BIOS 参数参考》、《华为服务器 Purley平台 BIOS 参数参考》和《华为服务器 Whitley平台 BIOS 参数参考》中的“常用任务 > 设置UEFI模式下Secure Boot功能”章节。

网卡固件升级的方法有多种：

- FwUpgrade方式：请参见对应版本的《SmartKit Computing x.x.x FwUpgrade 用户指南》。
- Smart Provisioning方式：请参见对应版本的《Smart Provisioning 用户指南》。
- SmartKit：请参见对应版本的《SmartKit Computing x.x.x 用户指南》。
- 命令行方式：对于命令行方式升级的指导，请参考本章节或对应服务器的升级指导书。

### 3.3.1 使用一键脚本升级固件

#### 前提条件

- 网卡驱动已安装。
- 已下载网卡的固件升级包。  
升级包的名称和获取方法可参见[3.1 获取软件包](#)。

#### 影响

升级过程中请勿下电，且除了升级指导内的操作，不要对其他配置进行修改。

##### 说明

当使用一键升级工具升级网卡固件时，若服务器下存在多个同型号的网卡，将自动升级所有同型号网卡固件至目标版本。

#### 操作步骤

**步骤1** 登录服务器iBMC远程虚拟控制台。

操作方法请参见[iBMC用户指南](#)。

**步骤2** 挂载升级文件。

##### 说明

iBMC远程虚拟控制台包含Java远程虚拟控制台和HTML5集成远程控制台，其中HTML5集成远程控制台不支持“本地文件夹”功能，本章节以Java远程虚拟控制台为例进行说明，更多信息可参考[iBMC用户指南](#)。

1. 在虚拟控制台的工具栏中单击，打开虚拟光驱下属菜单，如图3-5所示。

图 3-5 虚拟光驱



2. 选择“本地文件夹”并单击“浏览”，打开文件选择窗口。

3. 选择固件包解压后得到的文件夹。

- 如果执行版本升级操作，则此处挂载目标版本的升级包。
- 如果执行版本回退操作，则此处挂载升级前版本的升级包。

4. 单击“连接”。

当工具栏的“弹出”为可操作状态时，即表示已经成功挂载，如图3-6所示。

图 3-6 挂载成功



**步骤3** 拷贝升级文件到服务器，并为其赋予读写权限。

1. 执行`mount /dev/sr0 /mnt`命令，将本地文件夹挂载到“/mnt”路径下。

**说明**

可在输入`mount /dev/sr`后，按Tab键，自动联想起sr后面的值。

2. 执行`mkdir /root/fw`命令，在root目录下新建“fw”文件夹。
3. 执行`cp -vr /mnt/* /root/fw/`命令将所挂载的文件拷贝到“fw”目录下。
4. 执行`cd /root/fw/`命令进入“fw”目录。
5. 执行`chmod +x install.sh`为脚本添加可执行权限。

**步骤4** 执行命令使用固件一键升级工具升级固件。

`./install.sh upgrade`

命令执行时无回显打印，等待命令执行完成后，可执行以下命令查看升级情况。

- 执行`cat result.xml`命令查看升级结果，“Result”显示“OK”即表示升级成功。
- 执行`cat work.log`命令，打开所在文件夹的“work.log”日志文件查看升级日志信息。

**步骤5** 升级完毕后，将服务器下电再上电，使新的固件生效。

----结束

### 3.3.2 固件升级

**须知**

- 如IN200网卡的所有物理端口或PF均由虚拟机或Host DPDK接管，此时IN200不支持固件升级，可将IN200网卡任一物理端口或PF解除虚拟机或Host DPDK接管后再进行固件升级，固件升级完成后再按照实际业务场景使用虚拟机或Host DPDK接管对应网卡物理端口或PF。
- 升级网卡固件前，请查询“Secure Boot Status”参数值：
  - 若为“Disabled”，请直接升级。
  - 若为“Enabled”，请将“Secure Boot Status”参数设置为“Disabled”后再进行网卡升级，网卡升级完成后，再将参数设置为“Enabled”。

具体查询和设置“Secure Boot Status”参数值的方法，请参考《华为服务器Grantley平台 BIOS 参数参考》、《华为服务器Purley平台 BIOS 参数参考》和《华为服务器Whitley平台 BIOS 参数参考》中的“常用任务 > 设置UEFI模式下 Secure Boot功能”章节。

IN200固件升级的方法有多种：

IN200版本	支持的固件升级方式
SPC501及以下版本	命令行

IN200版本	支持的固件升级方式
SPC602及以上版本	<ul style="list-style-type: none"><li>● FwUpgrade</li><li>● Smart Provisioning</li><li>● 命令行</li></ul>

对于FwUpgrade和Smart Provisioning方式的升级指导，请参见对应服务器的升级指导书来操作。对于命令行方式升级的指导，请参考本章节。

## 前提条件

- IN200驱动已安装。
  - 已下载IN200网卡的固件升级包。  
升级包的名称和获取方法可参见[3.1 获取软件包](#)。
  - 升级包已上传到服务器OS。
  - hinicadm工具已安装。如果hinicadm工具版本为2.6.0.9及以上版本，则不做boot升级。

影响

升级过程不可中断，期间Linux或VMware ESXi不能重启，否则可能造成Linux或VMware ESXi异常、无法正常启动等情况。

## 操作步骤

- 步骤1 登录服务器OS。
  - 步骤2 进入升级包所在目录。
  - 步骤3 升级Firmware使之生效。

1. 执行

例如：

2. 执行**reboot**命令重启OS，使固件生效。

-----結束

## 后续操作

在固件升级完成后，您可以通过hinicadm工具的**hinicadm version -i <devicename>**命令查询生效固件的版本信息，确认是否已升级成功。其中<devicename>为IN200在操作系统中的设备名称，例如hinic0。

# 4 hinicadm 管理工具

hinicadm工具是专为IN200定制的管理工具。用户可以使用此工具对IN200进行管理。

[4.1 安装hinicadm](#)

[4.2 使用方法](#)

[4.3 命令列表](#)

[4.4 命令介绍](#)

[4.5 升级hinicadm](#)

[4.6 卸载hinicadm](#)

## 4.1 安装 hinicadm

### 前提条件

- 客户端已下载hinicadm工具的安装包。  
安装包的名称和获取方法可参见[3.1 获取软件包](#)。
- 安装包已上传到服务器OS。

### 在 Linux 系统下安装 hinicadm

**步骤1** 登录服务器OS。

**步骤2** 安装hinicadm工具。

- rpm包的安装：

执行**rpm -ivh hinicadm-xxxx-xxxx.xxx.rpm**。

例如：

```
linux-uueJIR:/Hi1822/tools/linux # rpm -ivh hinicadm-1.2.8.0-1.x86_64.rpm
Preparing... ################################ [100%]
Updating / installing...
1:hinicadm-1.2.8.0-1 ################################ [100%]
```

- deb包的安装：

执行**dpkg -i hinicadm-xxxx-xxxx.xxx.deb**。

例如：

```
root@ubuntu1804:/home/ubuntu # dpkg -i hinicadm-1.6.1.1-4.15.0_20_generic.arm64.deb
Selecting previously unselected package hinic.
(Reading database ... 72646 files and directories currently installed.)
Preparing to unpack hinicadm-1.6.1.1-4.15.0_20_generic.arm64.deb ...
Unpacking hinic (1.6.1.1) ...
Setting up hinic (1.6.1.1) ...
Installing... Please wait for a moment.
Install hinicadm tool successfully.
```

----结束

## 4.2 使用方法

### 命令格式

hinicadm工具提供的命令的格式为：

**hinicadm <major cmd> <minor cmd> <parameter> …<minor cmd> <parameter>**

一条命令的具体功能依靠主命令（**major cmd**）和子命令（**minor cmd**）来确定。

### 获取帮助信息

在使用hinicadm工具时，您可以使用如下几个常用的帮忙命令，帮助您更方便地使用此工具。

- 获取工具版本信息  
短命令：hinicadm -v  
长命令：hinicadm --version
- 获取工具支持的主命令列表  
短命令：hinicadm -h  
长命令：hinicadm --help
- 获取工具支持的子命令列表  
短命令：hinicadm <major cmd> -h  
长命令：hinicadm <major cmd> --help

此外，您可以在任意一个主命令后加-h，从而获取从属该主命令的所有子命令及其参数的取值范围。

## 4.3 命令列表

表4-1列出了hinicadm工具支持的主命令及其功能。

表 4-1 hinicadm 工具主命令

主命令	功能	备注
<a href="#">autoneg</a>	查询/设置自协商模式。	-
<a href="#">bmc_show</a>	查询BMC与IN200的带外交互信息。	-

主命令	功能	备注
<b>bp</b>	查询芯片反压信息。	<p>不支持以下操作系统的ARM版本：</p> <ul style="list-style-type: none"><li>• SLES 15/12.3/12.4</li><li>• RHEL 7.4/7.5/7.6</li><li>• CentOS 7.4/7.5/7.6</li><li>• EulerOS V2.0 SP8 ARM</li><li>• NeoKylin Server V5.0U5/V7.0U5/V7.0U6</li></ul> <p><b>说明</b></p> <ul style="list-style-type: none"><li>• 此命令仅用于开发人员问题定位，不会影响用户使用hinicadm工具。</li><li>• 在SR-IOV场景下，需要开启iommu或Smmu，同时需要配置passthrough，此命令才能使用。具体操作请参见<a href="#">6.2.1.2 修改系统引导配置文件</a>或<a href="#">6.2.2.2 修改系统引导配置文件</a>。</li></ul>
<b>clear</b>	清除统计信息。	-
<b>counter</b>	统计计数。	-

主命令	功能	备注
<a href="#">cpb</a>	获取芯片cpb cell使用情况。	不支持以下操作系统的ARM版本： <ul style="list-style-type: none"><li>• SLES 15/12.3/12.4</li><li>• RHEL 7.4/7.5/7.6</li><li>• CentOS 7.4/7.5/7.6</li><li>• EulerOS V2.0 SP8 ARM</li><li>• NeoKylin Server V5.0U5/V7.0U5/V7.0U6</li></ul>
<a href="#">csr_info</a>	查询芯片内部模块信息并保存。	<b>说明</b> <ul style="list-style-type: none"><li>• 此命令仅用于开发人员问题定位，不会影响用户使用hinicadm工具。</li><li>• 在SR-IOV场景下，需要开启iommu或Smmu，同时需要配置passthrough，此命令才能使用。具体操作请参见<a href="#">6.2.1.2 修改系统引导配置文件</a>或<a href="#">6.2.2.2 修改系统引导配置文件</a>。</li></ul>
<a href="#">currenet_port_traffic</a>	在currenet场景下查询RoCE各端口发送接收实时流量。	-
<a href="#">currenet_cc_param</a>	在currenet场景下查询和设置拥塞控制算法的参数。	-
<a href="#">dcb</a>	查询和设置DCB功能。	目前仅支持Linux系统下配置QoS使用。

主命令	功能	备注
<b>dp</b>	查询芯片数据通路相关模块统计。	不支持以下操作系统的ARM版本： <ul style="list-style-type: none"><li>• SLES 15/12.3/12.4</li><li>• RHEL 7.4/7.5/7.6</li><li>• CentOS 7.4/7.5/7.6</li><li>• EulerOS V2.0 SP8 ARM</li><li>• NeoKylin Server V5.0U5/V7.0U5/V7.0U6</li></ul> <b>说明</b> <ul style="list-style-type: none"><li>• 此命令仅用于开发人员问题定位，不会影响用户使用hinicadm工具。</li><li>• 在SR-IOV场景下，需要开启iommu或Smmu，同时需要配置passthrough，此命令才能使用。具体操作请参见<a href="#">6.2.1.2 修改系统引导配置文件</a>或<a href="#">6.2.2.2 修改系统引导配置文件</a>。</li></ul>
<b>ets</b>	查询和设置ETS功能。	目前仅支持Linux系统下配置QoS使用。
<b>event</b>	查询异步事件统计信息。	-
<b>fe_epc</b>	FE ( Fusion Engine ) 异常信息获取。	-
<b>fec</b>	查询/设置设备指定端口的FEC模式。	-
<b>filter</b>	查询和设置过滤功能。	-
<b>fm_show</b>	查询硬件模块上报的中断信息。	-
<b>hilink_count</b>	查询hilink统计。	-
<b>hilink_dump</b>	读取hilink寄存器信息。	-
<b>hilink_follow</b>	查询和设置链路跟随状态。	-
<b>hilink_param</b>	查询hi16/hi30参数。	-
<b>hilink_port</b>	查询端口信息。	-
<b>hilink_speed</b>	查询和设置端口速率。	-

主命令	功能	备注
<b>info</b>	查询系统设备列表或指定设备基本信息。	-
<b>log</b>	获取在线日志、解析离线日志。	-
<b>lro</b>	查询和设置lro聚合参数。	-
<b>mac</b>	查询MAC地址。	-
<b>mig_dirty_bitmap</b>	查询热迁移指定虚拟机ID的脏页信息。	-
<b>mig_info</b>	查询热迁移驱动的基本信息。	-
<b>mig_queue</b>	查询热迁移驱动的队列信息。	-
<b>mig_vf</b>	查询热迁移驱动vf的基本信息或者统计参数。	-
<b>mode</b>	查询/设置设备工作模式。	-
<b>nic_queue</b>	查询设备rx/tx队列信息。	-
<b>ovs_info</b>	查询OVS卸载信息	-
<b>pdm</b>	查询CPB CELL资源分配以及当前资源占用情况。	<p>不支持以下操作系统的ARM版本：</p> <ul style="list-style-type: none"><li>• SLES 15/12.3/12.4</li><li>• RHEL 7.4/7.5/7.6</li><li>• CentOS 7.4/7.5/7.6</li><li>• EulerOS V2.0 SP8 ARM</li><li>• NeoKylin Server V5.0U5/V7.0U5/V7.0U6</li></ul> <p><b>说明</b></p> <ul style="list-style-type: none"><li>• 此命令仅用于开发人员问题定位，不会影响用户使用hinicadm工具。</li><li>• 在SR-IOV场景下，需要开启iommu或Smmu，同时需要配置passthrough，此命令才能使用。具体操作请参见<a href="#">6.2.1.2 修改系统引导配置文件</a>或<a href="#">6.2.2.2 修改系统引导配置文件</a>。</li></ul>
<b>pfc</b>	查询和设置PFC功能。	目前仅支持Linux系统下配置QoS使用。

主命令	功能	备注
<b>qos</b>	查询/设置用户优先级、cqe水线。	仅支持Linux系统。
<b>rate</b>	查询/设置端口限速。	-
<b>reboot_notice</b>	设置和触发网卡在AC掉电时待发送的报文数据。	仅支持linux系统ARM版本。 IN200卡所有型号均可设置该参数，但仅当TaiShan 200服务器（型号 2280）配置SP580时有效。
<b>reset</b>	恢复出厂设置。	-
<b>roce_dscp</b>	配置ROCE_DSCP功能。	-
<b>roce_port_traffic</b>	查询roce设备端口发送接收实时流量。	-
<b>sdi_cfg</b>	查询和设置SDI配置。	仅支持Linux系统。 支持在OVS（Open vSwitch）卸载模式下查询和配置限速功能和管理VLAN功能；单独重启功能仅支持SD100卡。
<b>sdi_mode</b>	查询和设置SDI卡模式。	仅支持Linux系统。 仅支持SD100卡。
<b>self_adaption</b>	查询/设置端口协商自适应模式。	-
<b>serdes</b>	查询serdes信息。	-
<b>sfp</b>	查询光模块信息。	-
<b>sriov</b>	查询/设置pf虚拟化控制。	-
<b>table</b>	打印线性表信息。	-
<b>temperature</b>	查询芯片温度和光模块温度。	-

主命令	功能	备注
<a href="#">tile_io</a>	查询微码线程状态。	<p>不支持以下操作系统的ARM版本：</p> <ul style="list-style-type: none"><li>• SLES 15/12.3/12.4</li><li>• RHEL 7.4/7.5/7.6</li><li>• CentOS 7.4/7.5/7.6</li><li>• EulerOS V2.0 SP8 ARM</li><li>• NeoKylin Server V5.0U5/V7.0U5/V7.0U6</li></ul> <p><b>说明</b></p> <ul style="list-style-type: none"><li>• 此命令仅用于开发人员问题定位，不会影响用户使用hinicadm工具。</li><li>• 在SR-IOV场景下，需要开启iommu或Smmu，同时需要配置passthrough，此命令才能使用。具体操作请参见<a href="#">6.2.1.2 修改系统引导配置文件</a>或<a href="#">6.2.2.2 修改系统引导配置文件</a>。</li></ul>
<a href="#">updatefw</a>	升级固件。	-
<a href="#">version</a>	查询版本信息。	-
<a href="#">xsfp</a>	管理SFP/QSFP的信息和状态。	-

## 4.4 命令介绍

此处介绍hinicadm工具的常用命令的含义、参数及使用方法等。关于hinicadm命令的更多信息，请在工具中执行-h获取。

### 4.4.1 查询和设置指定设备的自协商模式 ( autoneg )

#### 命令功能

**autoneg**命令用于查询和设置指定设备的端口自协商模式。

#### 命令格式

**hinicadm autoneg -i <devicename> -p <portid>**

**hinicadm autoneg -i <devicename> -p <portid> -m <mode>**

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询和设置的IN200在系统中的名称	例如：hinic0、hinic1
<i>portid</i>	待查询和配置的端口号	-
<i>mode</i>	待设置的自协商模式	<ul style="list-style-type: none"><li>• 25g：表示采用25G联盟协议定义的自协商模式</li><li>• ieee：表示采用IEEE协议模式定义的自协商模式</li></ul> 两种模式的详细差异，请参见 IEEE802.3 和 25G Ethernet Consortium协议中的描述。

## 使用指南

无

## 使用实例

```
# 设置IN200 port1的自协商模式为“ieee”。
[root@localhost tool]# hinicadm autoneg -i hinic0 -p 1 -m ieee
Set port1 auto-negotiation mode to ieee mode succeed.

# 查询IN200 port1的自协商模式。
[root@localhost tool]# hinicadm autoneg -i hinic0 -p 1
Port1 auto-negotiation mode: ieee
```

## 4.4.2 查询 BMC 与 IN200 的带外交互信息 ( bmc\_show )

### 命令功能

**bpm**命令用于查询BMC与IN200的带外交互信息。

### 命令格式

```
hinicadm bmc_show -i <devicename> -m <mode>
hinicadm bmc_show -i <devicename> -m <mode> -p <portid> -t <type>
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1

参数	说明	取值
<i>mode</i>	BMC带外模式	<ul style="list-style-type: none"><li>ncsi</li><li>smbus</li><li>mctp</li></ul>
<i>portid</i>	待查询的端口号	-
<i>type</i>	待查询的ncsi的配置模式	<ul style="list-style-type: none"><li>0: ram, 运行过程中使用的配置。</li><li>1: flash, 固化的配置。</li></ul>

## 使用指南

- 查询smbus和mctp时，命令格式为**hinicadm bmc\_show -i <devicename> -m <mode>**。
- 查询ncsi时，命令格式为**hinicadm bmc\_show -i <devicename> -m <mode> -p <portid> -t <type>**。该命令可查询NC-SI的配置信息，并不代表IN200支持NC-SI相关功能。

## 使用实例

# 查询BMC与IN200的smbus交互信息。

```
linux-XapaGb:~ # hinicadm bmc_show -i hinic0 -m smbus
=====
[SBUS]slave address: 0x74
[SBUS]board type: 0x5
[SBUS]has sfp: 0x0 (0=yes)
[SBUS]work mode: 0x0
[SBUS]self check: 0x1 (1-ok)
[SBUS]core temperature: 0x33
[SBUS]health status: 0x0 (0-normal)
[SFP]port num: 0x4
[SFP]port identifier: 0x3 0x3 0x0 0x0
[SFP]link status: 0x1 0x1 0x0 0x0
[SFP]port speed: 0x19 0x19 0x0 0x0
[SFP]sfp temperature: 0x20 0x23 0x7ffe 0x7ffe
[BMC]opc_ability req:0x1 resp:0x1
[BMC]opc_core_temp req:0x2b resp:0x2b
[BMC]opc_sfp_temp req:0x2b resp:0x2b
[BMC]opc_phy_temp req:0x0 resp:0x0
[BMC]opc_firm_ver req:0x1 resp:0x1
[BMC]opc_inspec req:0x80 resp:0x80
[BMC]opc_link_status req:0x2b resp:0x2b
[BMC]opc_mac_addr req:0x2b resp:0x2b
[BMC]opc_dev_health req:0x56 resp:0x56
[BMC]opc_err_code req:0x0 resp:0x0
-----Mac Address-----
[MAC]pf(0x0) :xx:xx:xx:xx:xx:xx
[MAC]pf(0x1) :xx:xx:xx:xx:xx:xx
[MAC]pf(0x2) :xx:xx:xx:xx:xx:xx
[MAC]pf(0x3) :xx:xx:xx:xx:xx:xx
linux-XapaGb:~ # hinicadm bmc_show -i hinic0 -m mctp
=====
[MCTP]eid: 0x0
[MCTP]discovery: 0x0
[MCTP]lldp_enable: 0x0
[MCTP]curr_ctrl_cmd: 0x0
[MCTP]dest_ep_bdf: 0x800, 0x800, 0x0, 0x0
[MCTP]dest_ep_id: 0x61, 0x61, 0x61, 0x61
```

```
-----BMC-----
[BMC]get_link_status req:0x0 resp:0x0
[BMC]get_version_id req:0x0 resp:0x0
[BMC]get_capabilities req:0x0 resp:0x0
[BMC]get_parameters req:0x0 resp:0x0
[BMC]get_controller_packet_statistics req:0x0 resp:0x0
[BMC]get_ncsi_statistics req:0x0 resp:0x0
[BMC]get_ncsi_passthrough_statistics req:0x0 resp:0x0
[BMC]get_channel_state req:0x0 resp:0x0
[BMC]get_register_value req:0x0 resp:0x0
[BMC]oem_command req:0x0 resp:0x0
[BMC]oem_get_network_interface_bdf req:0x0 resp:0x0
[BMC]oem_get_pcie_ability req:0x0 resp:0x0
[BMC]oem_get_pcie_status req:0x0 resp:0x0
[BMC]oem_get_network_interface_media_type req:0x0 resp:0x0
[BMC]oem_get_junction_temp req:0x0 resp:0x0
[BMC]oem_get_optical_module_temp req:0x0 resp:0x0
[BMC]oem_get_err_code req:0x0 resp:0x0
[BMC]oem_get_network_interface_trans_cable_info req:0x0 resp:0x0
[BMC]oem_get_enable_lldp_capture req:0x0 resp:0x0
[BMC]oem_get_lldp_capbility req:0x0 resp:0x0
[BMC]oem_get_hw_ncsi_oem_cmd_capability req:0x0 resp:0x0
[BMC]oem_get_network_interface_mac_addr req:0x0 resp:0x0
[BMC]oem_get_network_interface_dcbx req:0x0 resp:0x0
[BMC]oem_get_network_default_mac_addr req:0x0 resp:0x0
[BMC]oem_get_enable_lldp_over_ncsi req:0x0 resp:0x0
[BMC]oem_get_lldp_over_ncsi_status req:0x0 resp:0x0
[BMC]oem_get_volatile_mac req:0x0 resp:0x0
```

# 查询BMC与IN200的ncsi交互信息，包括NC-SI线缆在位状态、NC-SI版本、NC-SI网口MAC信息等。

```
linux-XapaGb:~ # hinicadm bmc_show -i hinic0 -m ncsi -t 0 -p 0
=====
=====ncsi settings ram=====
ncsi_cable_state = 0x0
ncsi_ver = 0x01
ncsi_pkg_id = 0x0
arb_en = 0x1
duplex_set = 0x1
chan_num = 0x1
iid = 0x0
lldp_over_ncsi_enable = 0x0
lldp_over_mctp_enable = 0x0
magic_word = 0xdeadbeef
-----channel 0-----
aen_en = 0x0
index = 0x0
port = 0x0
state = 0x0
ncsi_port_en = 0x0
-----capabilities-----
capa_flags = 0x1d
bcast_filter = 0x1
multicast_filter = 0x1
buffering = 0x2ee0
aen_ctrl = 0x3
vlan_count = 0x8
mixed_count = 0x4
multicast_count = 0x4
unicast_count = 0x4
vlan_mode = 0x5
chan_count = 0x4
-----parameters-----
mac_address_count = 0x4
mac_address_flags = 0x0
vlan_tag_count = 0x8
vlan_tag_flags = 0x0
link_settings = 0x0
broadcast_packet_filter_settings = 0x0
channel_enable = 0x0
```

```
broadcast_packet_filter_status = 0x0
channel_network_tx_enable = 0x0
global_multicast_packet_filter_status = 0x0
vlan_mode = 0x0
flow_control_enable = 0x0
AEN_control = 0x0
mac_add[0]: xx.xx.xx.xx.xx.xx
mac_add[1]: xx.xx.xx.xx.xx.xx
mac_add[2]: xx.xx.xx.xx.xx.xx
mac_add[3]: xx.xx.xx.xx.xx.xx
vlan_tag: 0-0-0-0-0-0-0-0
```

## 4.4.3 查询芯片反压信息 ( bp )

### 命令功能

**bp**命令用于查询IN200芯片的反压信息。

### 命令格式

```
hinicadm bp -i <devicename>
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1

### 使用指南

无

### 使用实例

```
# 查询IN200的芯片反压信息。
[root@localhost tool]#hinicadm bp -i hinic0
===[ Hi1822 backpressure info]==
***** Hi1822 TX bp history counter *****
*** TX link level ***

*** TX sub level ***

***** Hi1822 RX bp history counter *****
*** RX link level bp ***

***** Hi1822 TX bp status *****
*** TX link level bp ***

*** TX sub_level bp ***

***** Hi1822 RX bp status *****
*** RX link level bp ***
```

```
*** RX sub_level bp ***
```

#### 4.4.4 清除指定设备统计信息 ( clear )

##### 命令功能

clear令用于清除指定设备的指定类型的统计信息。

##### 命令格式

```
hinicadm clear -i <nicdevicename> -t <type>
```

##### 参数说明

参数	说明	取值
<i>nicdevicename</i>	待清除统计信息的IN200在系统中的名称	例如: hinic0、hinic1  说明 <ul style="list-style-type: none"><li>-t取值为0或2时，可输入eth0。</li><li>清除驱动统计时，对象名称如果使用设备名（如hinic0），则清除hinic0上所有网络设备的对应统计，对象名称如果使用网络设备名（如eth0），则清除指定网络设备(eth0)上的统计。</li></ul>
<i>type</i>	待清除的统计信息类型	<ul style="list-style-type: none"><li>0: 表示异步信息统计</li><li>1: 表示DFX信息统计</li><li>2: 表示基本IO信息统计</li></ul>

##### 使用指南

当type=2时，设备名必须是网络端口名。

##### 使用实例

```
# 清除IN200的异步计数统计信息。
```

```
[root@localhost]# hinicadm clear -i hinic0 -t 0
Clear event stats succeed.
```

```
# 清除IN200的eth0的IO统计信息。
```

```
[root@localhost]# hinicadm clear -i eth0 -t 2
Clear driver stats succeed.
Clear vport stats succeed.
Clear port stats succeed.
```

## 4.4.5 查询统计信息 ( counter )

### 4.4.5.1 查询指定设备统计信息 ( -t -x )

#### 命令功能

**counter**命令用于查询指定设备的所有统计信息，包括MIB、微码、Firmware、IPSU、驱动、PCIe MAC层错误的统计信息。

**counter -t**命令用于查询指定设备指定类型的统计信息。

**counter -t -x**命令用于查询指定设备指定类型的指定序列的统计信息。

#### 命令格式

```
hinicadm counter -i <devicename>
```

```
hinicadm counter -i <devicename> -t <countertype>
```

```
hinicadm counter -i <devicename> -t <countertype> -x <counterindex>
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1 <b>说明</b> <ul style="list-style-type: none"><li>-t取值为3时，可输入eth0、eth1。</li><li>查询驱动统计时，对象名称如果使用hinic0则查询卡上的所有PF统计，对象名称如果使用网络设备名则查询指定网络设备的统计。</li></ul>
<i>countertype</i>	统计信息类型	<ul style="list-style-type: none"><li>0：表示Firmware统计信息</li><li>1：表示微码统计信息</li><li>2：表示IPSLURX统计信息</li><li>3：表示驱动统计信息</li><li>4：表示PCIe MAC层错误统计信息</li></ul>
<i>counterindex</i>	统计序列	可通过 <b>counter -h</b> 命令获取取值范围

#### 使用指南

当命令中包含-x时，“countertype”的取值范围为0、1。

#### 使用实例

```
# 查询IN200的所有统计信息。
```

```
[root@localhost tool]# hinicadm counter -i hinic0
uP Statistics:
```

```
uP Common Counter:  
mag module except: 117  
general level except: 246  
suggest level except: 127  
mag:err_mag_rf_lf: 252  
mag:err_mag_linkdown: 250  
mag:err_mag_linkup: 127  
.....
```

# 查询IN200的微码统计信息。

```
[root@localhost tool]# hinicadm counter -i hinic0 -t 1  
uCode Statistics:  
NIC MIB TX UC Counter:  
func1:tx_uc: 0x00000193b7694888 00000000197c70de  
func76:tx_uc: 0x0000000002e9c940 00000000000b4b76  
func77:tx_uc: 0x0000000ba3609e9c 00000000000bbd36  
NIC MIB TX BC Counter:  
func1:tx_bc: 0x0000000000000790 0000000000000002e  
func77:tx_bc: 0x00000000000002a 0000000000000001  
NIC MIB TX MC Counter:  
func0:tx_mc: 0x0000000000001930 00000000000000048  
func1:tx_mc: 0x000000000000ae3a 0000000000000001e3  
func2:tx_mc: 0x0000000000001944 00000000000000048  
func3:tx_mc: 0x0000000000001930 00000000000000048  
func76:tx_mc: 0x00000000000001b6 00000000000000005  
.....
```

# 查询IN200的微码类型的指定序列的统计信息。

```
[root@localhost tool]# hinicadm counter -i hinic0 -t 1 -x 1  
uCode Statistics:  
NIC MIB TX BC Counter:  
func1:tx_bc: 0x0000000000000790 0000000000000002e  
func77:tx_bc: 0x00000000000002a 0000000000000001  
.....
```

#### 4.4.5.2 离线解析巡检信息（-o）

##### 命令功能

**counter -o**命令用于离线解析巡检信息，并将解析结果自动保存到工具安装目录下。

##### 命令格式

```
hinicadm counter -o <countfile>
```

##### 参数说明

参数	说明	取值
<i>countfile</i>	待解析的巡检文件的文件名。 <b>说明</b> 该文件为通过iBMC一键信息收集功能获取到的IN200的带外巡检信息，其获取及存放路径，请参见《 <a href="#">iBMC 用户指南</a> 》。	例如： running_log.bin

##### 使用指南

执行此命令之前，请将待解析的巡检信息文件传输到工具的安装目录下。

## 使用实例

# 离线解析IN200的巡检信息。

```
[root@localhost tool]# hinicadm counter -o running_log_20180206095647.bin  
/opt/hinic/fwlog/inspection_info_bmc_offline_2018_02_07_10_22_38.log create succeed.
```

## 4.4.6 查询芯片 CPB Cell 使用情况 ( cpb )

### 命令功能

**cpb**命令用于查询IN200的芯片CPB ( Central Packet Buffer ) Cell使用情况。

### 命令格式

```
hinicadm cpb -i <devicename>
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1

### 使用指南

无

### 使用实例

# 查询IN200的芯片CPB cell使用情况。

```
[root@localhost tool]# hinicadm cpb -i hinic0  
***** CPB CELL NUM STATIC *****  
total_cell_num : 30720  
free_cell_num : 30720  
pdm_glb_num : 178  
cpi_octl_cell_num : 178  
leak_cell_num : 0  
fq_free_oeid_num : 511
```

## 4.4.7 查询芯片内部模块信息并保存 ( csr\_info )

### 命令功能

**csr\_info**命令用于查询芯片内部模块信息并保存。

### 命令格式

```
hinicadm csr_info -i <devicename> -m <module name> -t <type>
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1
<i>module name</i>	模块类型	cpi、cpb、ipsurx、ipsutx、qu、pe、esch、mqm、lcam、sml0、sml1、smf、tile0、tile1
<i>type</i>	寄存器类型	cfg、err、merr、int、cnt、ctp、cap、hst、mem、dummy <b>说明</b> module name为tile0或tile1时，不支持获取mem类型寄存器信息。

## 使用指南

- 使用此命令时不能同时运行业务。
- 不能针对同一张IN200执行两次此命令。

## 使用实例

```
# 查询hinic0芯片内部模块信息并保存。  
[root@localhost tool]# hinicadm csr_info -i hinic0 -m cpb -t err
```

## 4.4.8 在 currenet 场景下查询 RoCE 各端口发送接收实时流量 ( currenet\_port\_traffic )

### 命令功能

**currenet\_port\_traffic**命令用于在currenet场景下通过RoCE设备查询设备所在网卡各端口发送接收实时流量。

### 命令格式

```
hinicadm currenet_port_traffic -i <device> -t <exec_time>
```

## 参数说明

参数	说明	取值
<i>device</i>	有效RoCE设备名称	例如：hrn0_0、hrn0_bond_0
<i>exec_time</i>	执行时间，单位秒	-

## 使用指南

无

## 使用实例

```
# 在currenet场景下通过hrn0_bond_0实时显示所有RoCE端口流量状态，查看
hrn0_bond_0所在网卡所有端口的流量数据。
[root@10-113-213-141 ~]# hinicadm currenet_port_traffic -i hrn0_bond_0 -t 2

| port tx(Mpps) tx(Gbps) rx(Mpps) rx(Gbps) | port tx(Mpps) tx(Gbps) rx(Mpps) rx(Gbps) |
| 0 0.353 11.502978 0.309 10.054117 | 1 0.320 10.420641 0.297 9.668267 |
| 2 0.331 10.794715 0.290 9.452287 | 3 0.241 7.836618 0.290 9.421343 |
| port tx(Mpps) tx(Gbps) rx(Mpps) rx(Gbps) | port tx(Mpps) tx(Gbps) rx(Mpps) rx(Gbps) |
| 0 0.354 11.544683 0.301 9.808382 | 1 0.311 10.131188 0.302 9.834908 |
| 2 0.313 10.188379 0.291 9.496594 | 3 0.238 7.749376 0.298 9.707903 |
```

## 4.4.9 在 currenet 场景下查询和设置拥塞控制算法的参数 ( currenet\_cc\_param )

### 命令功能

**currenet\_cc\_param**命令用于在currenet场景下查询和设置拥塞控制算法的参数。

### 命令格式

```
hinicadm currenet_cc_param -i <devicename>
```

```
hinicadm currenet_cc_param -i <devicename> -s <value>
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1
<i>value</i>	算法参数	整形形式，0~255

### 使用指南

无

## 使用实例

#在currenet场景下设置hinic0拥塞控制算法的参数为255。

```
[root@localhost ~]# hinicadm currenet_cc_param -i hinic0 -s 255
```

#在currenet场景下查询hinic0拥塞控制算法的参数。

```
[root@localhost ~]# hinicadm currenet_cc_param -i hinic0
wnd_RST_ratio: 0xf
wnd_RST_tmout: 0xf
```

## 4.4.10 查询和设置 DCB 功能 ( dcb )

### 命令功能

**dcb**命令用于查询或设置设备的DCB功能。

### 命令格式

```
hinicadm dcb -i <devicename>
```

```
hinicadm dcb -i <devicename> -e <mode>
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待查询/设置的IN200在系统中的名称	例如: enp6s0、enp7s0
<i>mode</i>	控制DCB功能的开关模式	<ul style="list-style-type: none"><li>• 1: 开启DCB功能</li><li>• 0: 关闭DCB功能</li></ul>

### 使用指南

无

### 使用实例

```
# 查询enp6s0的DCB是否开启。
```

```
[root@localhost ~]# hinicadm dcb -i enp6s0
enp6s0    dcb state: disable
```

```
# 开启enp6s0的DCB。
```

```
[root@localhost ~]# hinicadm dcb -i enp6s0 -e 1
Set enp6s0 dcb to enable succeed.
```

## 4.4.11 查询芯片数据通路相关模块统计信息 ( dp )

### 命令功能

**dp**命令用于查询IN200芯片的数据通路相关模块的统计信息。

### 命令格式

```
hinicadm dp -i <devicename>
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1

## 使用指南

无

## 使用实例

# 查询IN200的芯片数据通路相关模块统计信息。

```
[root@localhost tool]#hinicadm dp -i hinic0
***** CPI GLOBAL STATIC *****
dwqe_api2sm    dwqe_dbe2mqm   nm_dbe2mqm   nm_dbe_drop
0              0             60            0

***** MQM statistics *****
rx_cpi_pkt     rx_sm_pkt    rx_tile_pkt   deq_iqm      deq_eqm    dis_cpi    filterd_db
60              0             0             60           0          0          0

age_drop       age2eqm      iqm2qu      iqm2sm
0              0             0             60
```

## 4.4.12 配置 ETS 功能 ( ets )

### □ 说明

设置ETS功能之前，首先需要使能DCB功能，具体操作请参见[4.4.10 查询和设置DCB功能 \( dcb \)](#)。

### 4.4.12.1 查询和设置 ETS 功能 ( -e )

#### 命令功能

**ets -e**命令用于查询或设置设备的ETS功能。

#### 命令格式

**hinicadm ets -i <devicename>**

**hinicadm ets -i <devicename> -e <mode>**

#### 参数说明

参数	说明	取值
<i>devicename</i>	待查询/设置的IN200在系统中的名称	例如：enp6s0、enp7s0

参数	说明	取值
<i>mode</i>	控制ETS功能的开关模式	<ul style="list-style-type: none"><li>• 1: 开启ETS</li><li>• 0: 关闭ETS</li></ul>

## 使用指南

无

## 使用实例

# 查询enp6s0的ETS是否开启。

```
[root@localhost ~]# hinicadm ets -i enp6s0
enp6s0      ets state: disable
up2tc:    3   3   3   3   3   2   1   0
pcnt:    100   0   0   0   0   0   0   0
strict:   0   0   0   0   0   0   0   0
```

# 开启enp6s0的ETS。

```
[root@localhost ~]# hinicadm ets -i enp6s0 -e 1
Set enp6s0 ets succeed
```

### 4.4.12.2 设置 SP 调度 ( -t )

#### 命令功能

**ets -t** 命令用于设置设备的SP ( strict priority ) 调度。

#### 命令格式

```
hinicadm ets -i <devicename> -e <mode> -t <tc>
```

```
hinicadm ets -i <devicename> -t <tc>
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如: enp6s0、enp7s0
<i>mode</i>	控制ETS功能的开关模式	<ul style="list-style-type: none"><li>• 1: 开启ETS</li><li>• 0: 关闭ETS</li></ul>
<i>tc</i>	控制流量到TC ( Traffic Classifier ) 之前的调度关系	xxxxxxxx x取值0~3

## 使用指南

- 当首次-e使能ETS后，再控制流量到TC之前的调度关系时，可以直接执行-t来实现。
- 当控制ETS功能的开关关闭后，只是不能配置ETS，控制流量到TC之前的调度关系并没有改变。ETS需要手动重新配置。

## 使用实例

```
# 设置enp6s0的优先级2、3的ETS为SP调度。
```

```
[root@localhost ~]# hinicadm ets -i enp6s0 -e 1 -t 00120000
Set enp6s0 ets succeed
```

### 4.4.12.3 设置 DWRR 调度 ( -t -p )

#### 命令功能

**ets -t -p**命令用于设置设备的DWRR调度。

#### 命令格式

```
hinicadm ets -i <devicename> -e <mode> [-t <tc>] -p <percent>
hinicadm ets -i <devicename> [-t <tc>] -p <percent>
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：enp6s0、enp7s0
<i>mode</i>	控制ETS功能的开关模式	<ul style="list-style-type: none"><li>• 1：开启ETS</li><li>• 0：关闭ETS</li></ul>
<i>tc</i>	控制流量到TC之前的调度关系	xxxxxxxx x取值0~3
<i>percent</i>	控制不同COS内的调度占用百分比	x,x,x,x,x,x,x x取值0~100

## 使用指南

- 当首次-e使能ETS后，再控制流量到TC之前的调度关系时，可以直接执行-t来实现。
- 当首次-e使能ETS后，再控制不同COS内的调度的占用的百分比时，可以直接执行-p来实现。
- 当控制ETS功能的开关关闭后，只是不能配置ETS，控制流量到TC之前的调度关系并没有改变。ETS需要手动重新配置。

- 当控制ETS功能的开关关闭后，只是不能配置ETS，控制不同COS内的调度占用百分比并没有改变。ETS需要手动重新配置。

## 使用实例

```
# enp6s0的优先级2、3的ETS为DWRR调度，优先级2带宽占比20%，优先级3带宽占比80%。
```

```
[root@localhost ~]# hinicadm ets -i <device> -e 1 -t 00120000 -p 0,0,20,80,0,0,0,0  
Set enp6s0 ets succeed
```

## 4.4.13 查询异步事件统计信息（event）

### 命令功能

event命令用于查询指定设备的异步事件统计信息。

### 命令格式

```
hinicadm event -i <devicename>
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1、eth0

### 使用指南

无

## 使用实例

```
# 查询IN200的异步事件统计信息。
```

```
[root@localhost tool]# hinicadm event -i hinic0  
SDK Statistics:  
  
Heartbeat Lost Counter:  
heartbeat lost: 0  
  
CQM Counter:  
cqm cmd alloc: 0  
cqm cmd free: 0  
cqm send cmd box: 0  
cqm send cmd imm: 0  
cqm db addr alloc: 0  
cqm db addr free: 0  
cqm fc srq create: 0  
cqm srq create: 0  
cqm rq create: 0  
cqm qpc mpt create: 0  
cqm nonrdma queue create: 0  
cqm rdma queue create: 0  
cqm rdma table create: 0  
cqm qpc mpt delete: 0
```

```
cqm nonrdma queue delete: 0
cqm rdma queue delete: 0
cqm rdma table delete: 0
cqm func timer clear: 0
cqm func hash buf clear: 0
cqm scq callback: 0
cqm ecq callback: 0
cqm nocq callback: 0

Link Event Counter:
link down: 0
link up: 0
pcie fault: 0

Chip Faults info:
module_id err_level er_type stats
0x02    0x01    0x6b    0x04
0x05    0x01    0x8c    0x04
```

## 4.4.14 查询异常记录 ( fe\_epc )

### 4.4.14.1 获取指定设备的异常信息 ( show )

#### 命令功能

**fe\_epc show**命令用于查看指定设备的FE异常信息。

#### 命令格式

```
hinicadm fe_epc -i <devicename> show -c <coreid> -t <tid> -n <num>
hinicadm fe_epc -i <devicename> show -b
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如： hinic0、 hinic1
<i>coreid</i>	待查询的coreid	-
<i>tid</i>	待查询的tid	-
<i>num</i>	显示FE异常信息条数	-

#### 使用指南

-b可显示所有异常记录。

#### 使用实例

# 显示关于IN200的历史FE异常信息。

```
[root@localhost tool]# hinicadm fe_epc -i hinic0 show -b
index: 192
```

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
vaild:0x0  
excp_code: 0x0  
core_id:0x0  
t_id:0x0  
I07_status:0x0  
I06_status:0x0  
I05_status:0x0  
I04_status:0x0  
I03_status:0x0  
I02_status:0x0  
I01_status:0x0  
I00_status:0x0  
.....
```

#### 4.4.14.2 触发指定 FE 产生并记录异常信息 ( -c -t )

##### 命令功能

**fe\_epc**命令可触发指定设备的FE异常信息。

##### 命令格式

```
hinicadm fe_epc -i <devicename> -c <coreid> -t <tid>  
hinicadm fe_epc -i <devicename> -a
```

##### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如： hinic0、 hinic1
<i>coreid</i>	待查询的coreid	-
<i>tid</i>	待查询的tid	-

##### 使用指南

带-a参数时，表示同时触发所有FE产生异常。

##### 使用实例

```
# 显示关于IN200所有FE产生的异常信息。
```

```
[root@localhost tool]# hinicadm fe_epc -i hinic0 -a  
index: 192  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
vaild:0x0  
excp_code: 0x0  
core_id:0x0  
t_id:0x0
```

```
I07_status:0x0
I06_status:0x0
I05_status:0x0
I04_status:0x0
I03_status:0x0
I02_status:0x0
I01_status:0x0
I00_status:0x0
.....
```

## 4.4.15 查询和设置指定端口 FEC 模式 ( fec )

### 4.4.15.1 查询端口 FEC 模式 ( -p )

#### 命令功能

**fec -p**命令用于查询设备指定端口的FEC ( Forward Error Correction ) 模式。

#### 命令格式

```
hinicadm fec -i <devicename> -p <portid>
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如: hinic0、hinic1
<i>portid</i>	待查询的端口ID	-

#### 使用指南

无

#### 使用实例

```
# 查询IN200的port0的FEC模式。
[root@localhost tool]# hinicadm fec -i hinic0 -p 0
Port0 FEC mode: nofec
```

### 4.4.15.2 设置指定端口 FEC 模式 ( -p -m )

#### 命令功能

**fec -p -m**命令用于设置设备指定端口的FEC模式。

#### 命令格式

```
hinicadm fec -i <devicename> -p <portid> -m <fecmode> [-d]
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如: hinic0、hinic1
<i>portid</i>	待设置的端口ID	-
<i>fecmode</i>	待设置的FEC模式	<ul style="list-style-type: none"><li>• rsfec: 表示RSFEC ( Reed Solomon Forward Error Correction ) 模式</li><li>• basefec: 表示基本的 FEC模式</li><li>• nofec: 表示不进行 FEC</li></ul>

## 使用指南

- 命令中带-d时，表示该设置为持久化配置，重启后依然有效。
- 命令中不带-d时，表示该设置非持久配置，在设备重启后失效，恢复为出厂配置。
- Fec模式设置并显示成功后，若当前光模块支持所设置的fec模式或者重新插上支持该fec模式的光模块时，fec模式才会生效。否则仍为原模式。

## 使用实例

```
# 设置IN200的port0的FEC模式为“rsfec”，且为持久化配置。
```

```
[root@localhost tool]# hinicadm fec -i hinic0 -p 0 -m rsfec -d
Set default FEC mode succeed.
Active default configuration succeed.
```

### 4.4.15.3 清除现有配置（-p -c）

#### 命令功能

**fec -p -c**命令用于清除现有配置，恢复FEC、自协商和强制速率的默认配置。

#### 命令格式

```
hinicadm fec -i <devicename> -p <portid> -c
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如: hinic0、hinic1

参数	说明	取值
<i>portid</i>	待设置的端口ID	-

## 使用指南

无

## 使用实例

```
# 清除现有配置，恢复FEC、自协商和强制速率的默认配置。
```

```
[root@localhost tool]# hinicadm fec -i hinic0 -p 0 -c
Clear default configuration succeed.
Active default configuration succeed.
```

## 4.4.16 查询和设置过滤功能 ( filter )

### 命令功能

filter命令用于查询和设置指定功能的过滤。

### 命令格式

```
hinicadm filter -i <devicename> -t <type>
hinicadm filter -i <devicename> -t <type> -e <mode>
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如： hinic0、 hinic1
<i>type</i>	功能类型，目前只支持源MAC为组播地址的检查功能	0: multicast smac check
<i>mode</i>	控制功能内容的开关模式	<ul style="list-style-type: none"><li>• 1: 开启</li><li>• 0: 关闭</li></ul>

## 使用指南

无

## 使用实例

```
# 查询hinic0的源MAC为组播地址的过滤是否开启。
```

```
[root@localhost ~]# hinicadm filter -i hinic0 -t 0
Now smac check: enable
```

# 开启hinic0的源MAC为组播地址的过滤功能。

```
[root@localhost ~]# hinicadm filter -i hinic0 -t 0 -e 1
Set smac check succeed.
```

#### 4.4.17 查询硬件模块上报的中断信息 ( fm\_show )

##### 命令功能

**fm\_show**命令用于查询硬件模块上报的中断信息。

##### 命令格式

```
hinicadm fm_show -i <devicename>
```

##### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hinic0、hinic1

##### 使用指南

无

##### 使用实例

# 显示hinic0硬件模块上报的中断信息。

```
[root@localhost ~]# hinicadm fm_show -i hinic0
*****no fm info!*****
```

#### 4.4.18 查询指定端口的统计信息 ( hilink\_count )

##### 命令功能

**hilink\_count**命令用于查询指定端口的统计信息。

##### 命令格式

```
hinicadm hilink_count -i <devicename> -p <portid>
```

##### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1
<i>portid</i>	待查询的端口号	-

## 使用指南

无

## 使用实例

# 查询port0的统计信息。

```
[root@localhost tool]# hinicadm hilink_count -i hinic0 -p 0
err_block = 0x00000000
base_fec = 0x00000000 0x00000000
rs_fec = 0x00000000 0x00000000 0x00000000
base_fec_idle = 0x00000000
MAG RXTP TXDP:
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
.......
```

## 4.4.19 查询自协商训练寄存器信息 ( **hilink\_dump** )

### 命令功能

**hilink\_dump**命令用于查询hi30模式下指定an\_train寄存器的配置信息。

### 命令格式

```
hinicadm hilink_dump -i <devicename> -p <portid>
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如： hinic0、 hinic1
<i>portid</i>	待查询的端口编号	-

## 使用指南

无

## 使用实例

# 查询port0端口的an\_train寄存器信息。

```
[root@localhost tool]# hinicadm hilink_dump -i hinic0 -p 0
an train:
port 0:
00000000 00000007 00000000 00000000 00000000 0020042b 00000021 00460023 ffffffff 0004c4b4
01e08c01 00000404 00000001 00000000 00000000 00000000
```

## 4.4.20 查询和设置链路跟随状态 ( hilink\_follow )

### 命令功能

hilink\_follow命令用于查询和设置链路跟随状态。

### 命令格式

```
hinicadm hilink_follow -i <devicename>
```

```
hinicadm hilink_follow -i <devicename> -s <state>
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待查询或设置的IN200在系统中的名称	例如: hinic0、eth0。
<i>state</i>	待设置的IN200的链路跟随状态	<ul style="list-style-type: none"><li>• default: 设置为当前环境默认的链路跟随状态。</li><li>• on: 设置IN200的链路跟随状态为开。</li><li>• off: 设置IN200的链路跟随状态为关。</li></ul>

### 使用指南

无

### 使用实例

```
# 查询hinic0的链路跟随状态。
```

```
[root@localhost hinic]# hinicadm hilink_follow -i hinic0
Now enp3s0 link follow off
Now enp4s0 link follow off
```

```
# 设置hinic0的链路跟随状态为开。
```

```
[root@localhost ~]# hinicadm hilink_follow -i hinic0 -s on
Set enp136s0 link follow on
Set enp138s0 link follow on
Set enp139s0 link follow on
Set enp137s0 link follow on
```

## 4.4.21 查询 hi16/hi30 参数 ( hilink\_param )

### 命令功能

hilink\_param命令用于查询指定设备在指定模式下的ctle、dfe、ffe参数。

## 命令格式

```
hinicadm hilink_param -i <devicename> -t<type> [-p <portid>]
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1
<i>type</i>	待查询的hinic类型	<ul style="list-style-type: none"><li>• 0: hi16信息</li><li>• 1: hi30信息</li></ul>
<i>portid</i>	待查询的端口号	<i>type</i> 为1时有效

## 使用指南

无

## 使用实例

# 查询hi16模式的参数信息。

```
[root@localhost tool]# hinicadm hilink_param -i hinic0 -t 0
HI16 parameter info:

lane 0 :
    TX_FFE: pre= 8;main= d;post= a
    RX_CTE: PASSGN=-2dB; ACTGN= 6 6 7; BST= 8 4 4; ZA= 1 1 1; SQH= 1 1 1; RMBAND= 1 1 1; CMBAND=
1 1 1;
    RX_DFE1: Tap1=0; Tap2=2; Tap3=2; Tap4=2; Tap5=2;
    RX_DFE2: Tap1=0; Tap2=0; Tap3=0; Tap4=0; Tap5=0; Tap6=0;
.....
```

## 4.4.22 查询指定端口的基本信息 ( hilink\_port )

### 命令功能

**hilink\_port**命令用于查询指定端口的基本信息。

### 命令格式

```
hinicadm hilink_port -i <devicename> -p <portid> [ -s ]
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1
<i>portid</i>	待查询的端口号	-

## 使用指南

- 不带-s参数时，可获取端口link相关所有信息，以及光模块信息。
- 带-s参数时，只能获取link状态、误码率等基本信息，不能获取光模块信息。

## 使用实例

# 查询port0的所有基本信息。

```
[root@localhost tool]# hinicadm hilink_port -i hinic1 -p 0
gpio_insert = present
link_state = link
an_state = off
alos = 0
rx_los = 0
speed = 25
fec = rsfec
rf_lf = 0x0
pma_fifo_reg = 0x000000020
pma_signal_ok_reg = 0x000000001
pcs_64_66b_reg = 0x000000000
pcs_err_cnt = 0x000000000
pcs_link = link
pma_ctrl = on
tx_enable = on
rx_enable = on
vendorName = Hisense
port_type = AOC
port_sub_type = NA
cable_length = 10(m)
cable_temperature = 41
max_speed = 25Gbps
sfp_type = sfp
tx_power = 737uW
rx_power = 716uW
his_link_machine_state = 0x9db70000
his_machine_state_length = 0
his_machine_state_data:
cur_link_machine_state = 0x9db70000
cur_machine_state_length = 0
cur_machine_state_data:
=====
port_id      lane_id      lane_num      parameter_cnt
0          0          1          6
-----
area |an      |speed      |fec      |used      |hi30_mode      |an_fec      |an_speed      |an_pause
-----
Default |enable    25Gbps    nofec      used      25GE        rsfec      25GE_BASE_CR  ALL_SUPPORT
BIOS  |unset     unset      unset      used      25GE        rsfec      25GE_BASE_CR  ALL_SUPPORT
TOOL  |unset     25Gbps    unset      used      25GE        rsfec      25GE_BASE_CR  ALL_SUPPORT
Final  |disable   25Gbps    rsfec      used      25GE        rsfec      25GE_BASE_CR  ALL_SUPPORT
-----
area |status    |link_type  |max_speed  |default_fec |support_speed
-----
WIRE  |present   FIBER/AOC 25GE        rsfec      -25GE-
-----
area |flash_cfg |cur_cfg   |wire_node  |an_en      |speed      |fec
-----
ADAPT |unset     off       unknown    disable    undefined  rsfec
-----
```

### 4.4.23 查询和设置端口速率参数 ( hilink\_speed )

#### 4.4.23.1 查询指定端口的当前速率 ( -p )

##### 命令功能

**hilink\_speed -p**命令用于查询指定端口的当前速率配置信息。

##### 命令格式

**hinicadm hilink\_speed -i <devicename> -p <portid>**

##### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hinic0、hinic1
<i>portid</i>	待设置的端口ID	-

##### 使用指南

无

##### 使用实例

# 查询Port0的当前速率配置。

```
[root@localhost tool]# hinicadm hilink_speed -i hinic0 -p 0
Port0 auto-negotiation: on
Port0 force speed: 25GE
Port0 default GE mode: disable
Port0 default auto-negotiation: on
Port0 default force speed: 25GE
```

#### 4.4.23.2 设置指定端口的当前速率 ( -p -s )

##### 命令功能

**hilink\_speed -p -s**命令用于设置指定端口的当前速率。

##### 命令格式

**hinicadm hilink\_speed -i <devicename> -p <portid> -s <speed>**

##### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hinic0、hinic1
<i>portid</i>	待设置的端口ID	-

参数	说明	取值
<i>speed</i>	设置强制速率	<ul style="list-style-type: none"><li>SP570/SP580/SP582/SP586/SC381：设置为25或10GE。</li><li>SP572/SC371：设置为100或40GE。</li></ul>

## 使用指南

- 自协商模式为关闭状态时，可设置端口速率。
- 光模块不支持此命令指定速率。
- 命令中带-d时，表示该设置为持久化配置，重启后依然有效。
- 命令中不带-d时，表示该设置非持久配置，在设备重启后失效，恢复为出厂配置。

## 使用实例

# 设置Port0的速率为25GB。

```
[root@localhost ~]# hinicadm hilink_speed -i hinic0 -p 0 -an off -s 25
Set forced speed succeed.
The set rate will only take effect when auto-negotiation is turned off.
Set auto-negotiation succeed.
```

### 4.4.23.3 设置指定端口的自协商模式的开启状态（-p -an）

#### 命令功能

**hilink\_speed -p -an**命令用于设置指定端口自协商模式的开启状态。

#### 命令格式

```
hinicadm hilink_speed -i <devicename> -p <portid> -an <state> [-s <speed>] [-d]
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hinic0、hinic1
<i>portid</i>	待设置的端口ID	-
<i>state</i>	自协商模式的状态	<ul style="list-style-type: none"><li>on</li><li>off</li></ul>

参数	说明	取值
<i>speed</i>	设置强制速率	<ul style="list-style-type: none"><li>SP570/SP580/SP582/SP586/SC381：设置为25或10GE。</li><li>SP572/SC371：设置为100或40GE。</li></ul>

## 使用指南

- 自协商模式为关闭状态时，可设置端口速率。
- 光模块不支持此命令指定速率。
- 命令中带-s时，表示设置强制速率，需要关闭自协商，即-an的参数设置为off。
- 命令中带-d时，表示该设置为持久化配置，重启后依然有效。
- 命令中不带-d时，表示该设置非持久配置，在设备重启后失效，恢复为出厂配置。

## 使用实例

# 关闭Port0的自协商模式，且为持久配置。

```
[root@localhost ~]# hinicadm hilink_speed -i hinic0 -p 0 -an off -s 10 -d
Set default configuration succeed.
Active default configuration succeed.
```

### 4.4.23.4 设置指定端口的连接模式（-p -m）

#### 命令功能

**hilink\_speed -p -m**命令用于设置指定端口的连接模式。

#### 命令格式

```
hinicadm hilink_speed -i <devicename> -p <portid> -m <mode> [-d]
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hinic0、hinic1
<i>portid</i>	待设置的端口ID	-
<i>mode</i>	网口的连接模式。	<ul style="list-style-type: none"><li>enabled：表示速率模式为GE。</li><li>disabled：表示速率模式为10GE/25GE。</li></ul>

## 使用指南

- 命令中带-d时，表示该设置为持久化配置，重启后依然有效。
- 命令中不带-d时，表示该设置非持久配置，在设备重启后失效，恢复为出厂配置。
- 配置网卡的连接模式时，port0和port1联动，port2和port3联动。例如，将port0的连接模式由enabled修改为disabled并重启生效后，port1的连接模式自动变为disabled。
- GE模式仅用于调试场景，设置GE模式需要重启后生效，且设置为GE模式后，无法配置FEC、强制速率和自协商。

## 使用实例

```
# 使能port0端口的GE模式且持久生效。
```

```
[root@localhost ~]# hinicadm hilink_speed -i hinic0 -p 0 -m enable -d
Set default configuration succeed.
Active default configuration succeed.
```

### 4.4.23.5 清除指定端口的持久化速率配置（-p -c）

#### 命令功能

hilink\_speed -c命令用于清除指定端口的持久化速率配置，即清除[4.4.23.2 设置指定端口的当前速率（-p -s）](#)的-d属性。

#### 命令格式

```
hinicadm hilink_speed -i <devicename> -p <portid> -c
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hinic0、hinic1
<i>portid</i>	待设置的端口ID	-

## 使用指南

执行此命令后，对设备当前运行状态无影响；待系统重启后，端口的速率配置恢复到出厂设置。

## 使用实例

```
# 清除Port0的持久化配置。
```

```
[root@localhost tool]# hinicadm hilink_speed -i hinic0 -p 0 -c
Clear default configuration succeed.
Active default configuration succeed.
```

## 4.4.24 查询 IN200 基本信息 ( info )

### 命令功能

info命令用于查询服务器上全部IN200或指定IN200的基本信息。

### 命令格式

**hinicadm info**

**hinicadm info -i <devicename>**

### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1

### 使用指南

- FW或hinicadm工具任意一个组件版本为2.4.0.x之前版本，回显信息中无card name和flash name。
- FW和hinicadm工具都为2.4.0.x及之后版本，回显中有card name（比如：SP570），表示卡的名称，有flash name，表示flash型号。
- 若回显中card name或flash name显示为UNKNOWN，可通过升级最新版本的hinicadm工具查看准确信息。

### 使用实例

# 查询服务器上所有IN200的基本信息。

```
[root@localhost tool]# hinicadm info
Hi1822 device Information
  Card      PCle Function
  ---- hinic0(ETH)
  |----- 08:00.0 (NIC:eth5)
  |----- 08:00.1 (NIC:eth6)
  |----- 08:00.2 (NIC:eth7)
  |----- 08:00.3 (NIC:eth8)
  |--- hinic1(ETH)
  |----- 09:00.0 (NIC:eth9)
  |----- 09:00.1 (NIC:eth10)
  |----- 09:00.2 (NIC:eth11)
  |----- 09:00.3 (NIC:eth12)
```

# 查询指定IN200的基本信息。

```
[root@localhost tool]# hinicadm info -i hinic0
Card information:
card type    : ETH_DUAL_25GE(id=21)
card name    : SP582
port num     : 2
port speed   : 25GE
pcie width   : 8
host num     : 1
pf num       : 2
```

```
vf total num : 120
tile num : 2
qcm num : 6
core num : 4
work mode : 0
service mode : 2
pcie mode : X8_(id=8)
cfg addr : 0x60000
boot sel : 0
board id : 2
flash name : MX25U12835F
```

## 4.4.25 收集解析日志信息 ( log )

### 4.4.25.1 收集指定设备的所有固件日志信息 ( -a )

#### 命令功能

**log -a**命令用于收集指定设备的所有固件日志，包括RAM区的Firmware、微码日志，FLASH区的Firmware、微码日志，以及Firmware、微码的临终遗言等。

收集到的日志自动保存到/opt/hinic/fwlog/下。

#### 命令格式

```
hinicadm log -i <devicename> -a
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如： hinic0、 hinic1

#### 使用指南

无

#### 使用实例

# 查询指定IN200上的所有日志信息。

```
[root@localhost tool]# hinicadm log -i hinic0 -a
/opt/hinic/fwlog/ucode_flash_hinic0_2018_01_15_17_05_41.log create succeed.
/opt/hinic/fwlog/ucode_ram_hinic0_2018_01_15_17_05_41.log create succeed.
/opt/hinic/fwlog/ucode_lastword_flash_hinic0_2018_01_15_17_05_41.log create succeed.
/opt/hinic/fwlog/up_lastword_flash_hinic0_2018_01_15_17_05_41.log create succeed.
/opt/hinic/fwlog/up_flash_hinic0_2018_01_15_17_05_41.log create succeed.
/opt/hinic/fwlog/up_ram_hinic0_2018_01_15_17_05_40.log create succeed.

[root@localhost tool]# /opt/hinic/fwlog # ls
ucode_flash_hinic0_2018_01_15_17_05_41.log
ucode_ram_hinic0_2018_01_15_17_05_41.log
up_lastword_flash_hinic0_2018_01_15_17_05_41.log
ucode_lastword_flash_hinic0_2018_01_15_17_05_41.log
up_flash_hinic0_2018_01_15_17_05_41.log
up_ram_hinic0_2018_01_15_17_05_40.log
```

#### 4.4.25.2 收集指定类型的日志信息（ -t ）

##### 命令功能

**log -t**命令用于收集指定类型的日志信息，并将其自动保存在/opt/hinic/fwlog目录下。

##### 命令格式

```
hinicadm log -i <devicename> -t <logtype>
```

##### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如： hinic0、 hinic1
<i>logtype</i>	待收集的日志类型	<ul style="list-style-type: none"><li>• 0：表示RAM区的Firmware日志</li><li>• 1：表示FLASH区的Firmware日志</li><li>• 2：表示RAM区的微码日志</li><li>• 3：表示FLASH区的微码日志</li><li>• 4：表示Firmware和微码的临终遗言</li></ul>

##### 使用指南

无

##### 使用实例

```
# 查询指定IN200上的FLASH区Firmware日志。
```

```
[root@localhost tool]# hinicadm log -i hinic0 -t 1  
/opt/hinic/fwlog/ up_flash_hinic0_2018_01_15_17_05_41.log create succeed.
```

#### 4.4.25.3 离线解析运行日志信息（ -o1 ）

##### 命令功能

**log -o1**命令用于离线解析Firmware、微码运行日志，并将解析结果自动保存到/opt/hinic/fwlog目录下。

##### 命令格式

```
hinicadm log -o1 <logfile> -m <mgmt_index> -u <ucode_index>
```

## 参数说明

参数	说明	取值
<i>logfile</i>	待解析的日志文件的文件名。 <b>说明</b> 该文件为通过iBMC一键信息收集功能获取到的IN200的带外运行日志，其获取及存放路径，请参见《 <a href="#">iBMC 用户指南</a> 》。	例如： err_log.bin
<i>mgmt_index</i>	待解析的日志文件对应的Firmware索引文件。 <b>说明</b> 该文件的归档地址与下载方法与IN200的驱动、固件相同， 详细信息请参见 <a href="#">3.1 获取软件包</a> 。	例如： up_index
<i>ucode_index</i>	待解析的日志文件对应的微码索引文件。 <b>说明</b> 该文件的归档地址与下载方法与IN200的驱动、固件相同， 详细信息请参见 <a href="#">3.1 获取软件包</a> 。	例如： ucode_index

## 使用指南

执行此命令之前，请将待解析的日志文件、索引文件传输到工具的安装目录下。

执行此命令时，可选择-m、-u任意一个参数，也可两个参数同时存在。

## 使用实例

# 离线解析IN200的日志信息。

```
[root@localhost tool]# hinicadm log -o1 error_log.bin -m up_index -u ucode_index
/opt/hinic/fwlog/ up_ram_offline_2018_01_15_18_52_32.log create succeed.
/opt/hinic/fwlog/ up_flash_offline_2018_01_15_18_52_32.log create succeed.
/opt/hinic/fwlog/ ucode_ram_offline_2018_01_15_18_52_32.log create succeed.
/opt/hinic/fwlog/ ucode_flash_offline_2018_01_15_18_52_32.log create succeed.
[root@localhost tool]# /opt/hinic/fwlog # ls
/opt/hinic/fwlog/ up_ram_offline_2018_01_15_18_52_32.log create succeed.
/opt/hinic/fwlog/ up_flash_offline_2018_01_15_18_52_32.log create succeed.
/opt/hinic/fwlog/ ucode_ram_offline_2018_01_15_18_52_32.log create succeed.
/opt/hinic/fwlog/ ucode_flash_offline_2018_01_15_18_52_32.log create succeed.
```

### 4.4.25.4 离线解析临终遗言（-o2）

## 命令功能

**log -o2**命令用于离线解析Firmware、微码的临终遗言，并将解析结果自动保存到/opt/hinic/fwlog目录下。

## 命令格式

**hinicadm log -o2 <logfile>**

## 参数说明

参数	说明	取值
<i>logfile</i>	待解析的临终遗言文件的文件名。 <b>说明</b> 该文件为通过iBMC一键信息收集功能获取到的IN200的带外临终遗言，其获取及存放路径，请参见《 <a href="#">iBMC 用户指南</a> 》。	例如： last_word.bin

## 使用指南

执行此命令之前，请将待解析的临终遗言文件传输到工具的安装目录下。

## 使用实例

# 离线解析IN200的临终遗言。

```
[root@localhost tool]# hinicadm log -o2 last_word.bin
/opt/hinic/fwlog/ ucode_lastword_flash_offline_2018_01_15_18_58_25.log create succeed.
/opt/hinic/fwlog/ up_lastword_flash_offline_2018_01_15_18_58_25.log create succeed.
[root@localhost tool]# /opt/hinic/fwlog # ls
ucode_lastword_flash_offline_2018_01_15_18_58_25.log
up_lastword_flash_offline_2018_01_15_18_58_25.log
```

## 4.4.26 查询和设置 LRO 参数 ( lro )

### 4.4.26.1 查询整卡的 LRO 参数

#### 命令功能

**lro**命令用于查询整卡的LRO参数。

#### 命令格式

**hinicadm lro -i <devicename>**

#### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如： hinic0、 hinic1

## 使用指南

无

## 使用实例

# 查询整卡的LRO参数。

```
[root@localhost tool]# hinicadm lro -i hinic0
LRO coalesce time: 10us
function 0 LRO coalesce wqe number: 0
function 0 LRO ipv4 en: 0
function 0 LRO ipv6 en: 0
function 1 LRO coalesce wqe number: 8
function 1 LRO ipv4 en: 0
function 1 LRO ipv6 en: 0
function 2 LRO coalesce wqe number: 0
function 2 LRO ipv4 en: 0
function 2 LRO ipv6 en: 0
function 3 LRO coalesce wqe number: 0
function 3 LRO ipv4 en: 0
function 3 LRO ipv6 en: 0
```

#### 4.4.26.2 统一设置整卡所有 function 的 LRO 聚合个数 ( -n )

##### 命令功能

**lro -n** 命令用于统一设置整卡所有function的LRO聚合个数。

##### 命令格式

```
hinicadm lro -i <devicename> -n <num>
```

##### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如： hinic0、 hinic1
<i>num</i>	聚合个数。	整数形式： 1~32

##### 使用指南

无

##### 使用实例

```
# 统一设置整卡所有function的LRO合个数。
```

```
linux-4it5:~ # hinicadm lro -i hinic0 -n 32
Set function 0 LRO coalesce number to 32 wqe succeeded.
Set function 1 LRO coalesce number to 32 wqe succeeded.
Set function 2 LRO coalesce number to 32 wqe succeeded.
Set function 3 LRO coalesce number to 32 wqe succeeded.
```

#### 4.4.26.3 设置整卡的 LRO 聚合时间 ( -t )

##### 命令功能

**lro -t** 命令用于设置整卡的LRO聚合时间。

## 命令格式

```
hinicadm lro -i <devicename> -t <lro coalesce time>
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1
<i>lro coalesce time</i>	聚合时间。单位：微秒。	整数形式：1~1024

## 使用指南

无

## 使用实例

# 设置整卡的LRO聚合时间。

```
[root@localhost tool]# hinicadm lro -i hinic0 -t 100
Set LRO timer to 100us succeed.
```

### 4.4.26.4 查询单个 function 的 LRO 参数 ( -x )

## 命令功能

**lro -x**命令用于查询单个function的LRO参数。

## 命令格式

```
hinicadm lro -i <devicename> -x <index>
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1
<i>index</i>	function ID	整数形式：0~3, 16~255

## 使用指南

无

## 使用实例

# 查询单个function的LRO参数。

```
linux-4it5:~ # hinicadm lro -i hinic0 -x 0
LRO coalesce time: 16us
function 0 LRO coalesce wqe number: 0
function 0 LRO ipv4 en: 0
function 0 LRO ipv6 en: 0
```

#### 4.4.26.5 设置单个 function 的 LRO 聚合个数 ( -x -n )

##### 命令功能

**lro -x -n**命令用于设置单个function的LRO聚合个数。

##### 命令格式

```
hinicadm lro -i <devicename> -x <index> -n <num>
```

##### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如: hinic0、hinic1
<i>index</i>	function ID	整数形式: 0~3, 16~255
<i>num</i>	聚合个数	整数形式: 1~32

##### 使用指南

无

##### 使用实例

# 设置单个function的LRO聚合个数。

```
linux-4it5:~ # hinicadm lro -i hinic0 -x 0 -n 32
Set function 0 LRO coalesce number to 32 wqe succeeded.
```

#### 4.4.27 查询 MAC 地址 ( mac )

##### 命令功能

**mac**命令用于查询指定设备的MAC地址。

##### 命令格式

```
hinicadm mac -i <devicename>
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1

## 使用指南

无

## 使用实例

# 查询IN200的MAC地址信息。

```
[root@localhost tool]# hinicadm mac -i hinic0
mac[0]: ****:****:*****
mac[1]: ****:****:*****
mac[2]: ****:****:*****
mac[3]: ****:****:*****
mac[4]: ****:****:*****
mac[5]: ****:****:*****
mac[6]: ****:****:*****
mac[7]: ****:****:*****
mac[8]: ****:****:*****
mac[9]: ****:****:*****
mac[10]: ****:****:*****
mac[11]: ****:****:*****
mac[12]: ****:****:*****
mac[13]: ****:****:*****
mac[14]: ****:****:*****
mac[15]: ****:****:*****
```

## 4.4.28 查询热迁移指定虚拟机 ID 的脏页信息 ( mig\_dirty\_bitmap )

### 命令功能

**mig\_dirty\_bitmap**命令用于查询热迁移指定虚拟机ID的脏页信息。

### 命令格式

```
hinicadm mig_dirty_bitmap -i <devicename> -x <index>
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1
<i>index</i>	待查询的虚拟机的index	整数形式：0~3

## 使用指南

无

## 使用实例

```
# 查询IN200的vm_id为1的热迁移驱动的脏页信息。
```

```
linux-NdDUqe:~ # hinicadm mig_dirty_bitmap -i hinic0 -x 1
hinic0 migrate dirty buf information:
dom_uuid: 0
dirty_bitmap: (nil)
dirty_bitmap2: (nil)
block_nums: 0
remainder_page_order: 0
ref_cnt: 0
bitmap_in_use: 0
pci_name:
```

## 4.4.29 查询热迁移驱动的基本信息（mig\_info）

### 命令功能

**mig\_info**命令用于查询热迁移驱动的基本信息。

### 命令格式

```
hinicadm mig_info -i <devicename>
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1

## 使用指南

无

## 使用实例

```
# 查询IN200的热迁移驱动的基本信息。
```

```
linux-NdDUqe:~ # hinicadm mig_info -i hinic0
hinic0 migrate information:
total_qp_num: 32
used_qp_num: 0
total_dirty_bitmatp_num: 4
used_dirty_bitmatp_num: 0
total_vf_num: 128
```

## 4.4.30 查询热迁移驱动的队列信息 ( mig\_queue )

### 命令功能

**mig\_queue**命令用于查询热迁移驱动的队列信息。

### 命令格式

```
hinicadm mig_queue -i <ethdevicename> -d <direction> -t <type> -q <queueid>  
[-w <wqid>]
```

### 参数说明

参数	说明	取值
<i>ethdevicename</i>	待查询的网络设备在系统中的名称	例如：eth0、eth1
<i>direction</i>	待查询的队列方向	<ul style="list-style-type: none"><li>• 0：表示发送方向 ( tx )</li><li>• 1：表示接收方向 ( rx )</li></ul>
<i>type</i>	待查询的信息类型	0：表示队列信息 1：表示工作队列入口信息 2：表示CI表信息（仅支持tx） 3：表示CQE信息（仅支持rx）
<i>queueid</i>	待查询的队列ID	-
<i>wqid</i>	待查询的WQE ID	-

### 使用指南

无

### 使用实例

```
# 查询热迁移驱动的队列信息。
```

```
linux-NdDUqe:~ # hinicadm mig_queue -i hinic0 -t 1 -d 0 -q 0 -w 1023  
function 0 send queue ci table content:  
ci_addr:0x844284400  
pending_limit:0  
coalescing_time:0  
dma_attr_off:0  
int_en:1  
int_offset:3  
l2nic_sqn:0  
version:0  
status:0
```

## 4.4.31 查询热迁移驱动 vf 的基本信息或者统计参数 ( mig\_vf )

### 命令功能

**mig\_vf**命令用于查询热迁移驱动vf的基本信息或者统计参数。

### 命令格式

```
hinicadm mig_vf -i <devicename> -t <mig_vf_type> -x <pci_name>
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如: hinic0、hinic1
<i>mig_vf_type</i>	查询的热迁移驱动vf的类别	<ul style="list-style-type: none"><li>• vf_info: 表示基本信息</li><li>• vf_state: 表示统计</li></ul>
<i>pci_name</i>	查询的热迁移驱动vf的pci名称	例如0000:b5:00.0

### 使用指南

无

### 使用实例

# 查询热迁移驱动vf的基本信息。

```
linux-NdDUqe:~ # hinicadm mig_vf -i hinic0 -t vf_info -x 0000:b5:00.1
hinic0 migrate vf information:
fun_id: 16
num_qps: 8
num_qps_in_use: 8
vport_id: 1040
num_qps_max_allowed: 8
pf_id: 3
host_id: 2
cmdq_depth: 12
cqe_depth: 10
num_dma_attr: 0
cmdq_cla1_addr_h: 1
cmdq_cla1_addr_l: 876347392
cmdq_cla2_addr_h: 0
cmdq_cla2_addr_l: 3106144256
sq0_ci_addr_h: 0
sq0_ci_addr_l: 3134943233
mac: xx xx xx xx xx xx
```

# 查询热迁移驱动vf的统计参数。

```
linux-NdDUqe:~ # hinicadm mig_vf -i hinic0 -t vf_state -x 0000:b5:00.1
hinic0 migrate vf state:
migrate vf dirty bitmap:
dom_uuid: 0
dirty_bitmap: (nil)
```

```
dirty_bitmap2: (nil)
block_nums: 0
remainder_page_order: 0
ref_cnt: 0
bitmap_in_use: 0
migrate vf log info:
cmd: 0
iter: 0
dom_uuid: 3074956339
offset: 0
buf_size: 163848
segment: 0
migrate vf state:
master_vf: 1
is_deactivated: 1
migrating: 0
log_stop: 0
time_out: 0
fail_stage: 1
is_vport_valid: 0
end_time: 5224928141
cqe_gpa: 0xfffff891408c53200
recv_gpa: 0
dfx_count: 0
func_id: 16
q_id: 0
mbox_cmdq_busy: 0
```

## 4.4.32 查询和设置指定设备的工作模式 ( mode )

### 命令功能

mode命令用于查询和设置指定设备的工作模式。

### 命令格式

```
hinicadm mode -i <devicename>
hinicadm mode -i <devicename> -m <workmode>
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待查询和设置的IN200在系统中的名称	例如： hinic0、 hinic1
<i>workmode</i>	待设置的工作模式	<ul style="list-style-type: none"><li>• nic： 表示标准NIC工作模式。</li><li>• ovs： 表示OVS ( Open Virtual Switch ) 卸载工作模式。</li></ul>

### 使用指南

配置后需重启OS生效。

### 使用实例

```
# 设置IN200的工作模式为“ovs”。
```

```
[root@localhost tool]# hinicadm mode -i hinic0 -m ovs
Set work mode to ovs succeed.
Please reboot OS for the new mode to take effect.
```

# 查询IN200的工作模式。

```
[root@localhost tool]# hinicadm mode -i hinic0
Current work mode: NIC
Work mode in permanent configuration: OVS
```

## 4.4.33 查询指定设备队列信息（nic\_queue）

### 命令功能

**nic\_queue**命令用于查询指定网络设备的队列信息。

### 命令格式

```
hinicadm nic_queue -i <ethdevicename> -d <direction> -t <type> -q <queueid> [-w <wqid>]
```

### 参数说明

参数	说明	取值
<i>ethdevicename</i>	待查询的网络设备在系统中的名称	例如：eth0、eth1
<i>direction</i>	待查询的队列方向	<ul style="list-style-type: none"><li>• 0：表示发送方向（tx）</li><li>• 1：表示接收方向（rx）</li></ul>
<i>type</i>	待查询的信息类型	0：表示队列信息 1：表示工作队列入口信息 2：表示CI表信息（仅支持tx） 3：表示CQE信息（仅支持rx）
<i>queueid</i>	待查询的队列ID	-
<i>wqid</i>	待查询的WQE ID	-

### 使用指南

无

### 使用实例

# 查询IN200的eth0的队列信息。

```
[root@localhost tool]# hinicadm nic_queue -i eth0 -d 1 -t 0 -q 0
Recieve queue0 information:
```

```
queue id:0
hw_pi:1023
ci:5
sw_pi:1023
rq_depth:1024
rq_wqebb_size:32
.....
```

#### 4.4.34 查询 OVS 卸载信息 ( ovs\_info )

##### 命令功能

**ovs\_info**命令用于查询指定IN200设备的OVS ( Open vSwitch ) 卸载信息。

##### 命令格式

```
hinicadm ovs_info -i <devicename> -t <type> -x <index>
```

##### 参数说明

参数	说明	取值
<i>ethdevicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1。
<i>type</i>	待查询设备的OVS卸载信息的类型	<ul style="list-style-type: none"><li>• fm_bheap：表示流监控使用的bitmap信息。</li><li>• fm_counter：表示流监控统计信息。</li></ul>
<i>index</i>	待查询设备的流监控索引	整数形式，0~2303。

##### 使用指南

无

##### 使用实例

```
# 查询hinic0的流监控使用的bitmap信息。
```

```
[root@localhost ~]# hinicadm ovs_info -i hinic0 -t fm_bheap -x 0
fm_bheap:latency_id(0) bheap(0)
```

```
# 查询hinic0流监控的统计信息。
```

```
[root@localhost ~]# hinicadm ovs_info -i hinic0 -t fm_counter -x 0
fm_bheap:latency_id(0) counter(0x0 0x0)
```

#### 4.4.35 查询 CPB Cell 资源分配以及当前资源占用情况 ( pdm )

#### 4.4.35.1 查询某个模块下的 CPB Cell 资源分配以及当前资源占用情况 ( -m )

##### 命令功能

pdm -m命令用于查询某个模块下的CPB Cell资源分配以及当前资源占用情况。

##### 命令格式

**hinicadm pdm -i <devicename> -m <module name>**

##### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hnic0、hnic1
<i>module name</i>	指定显示资源类型。	all、glb、pg、tx、rx、yzone、rzone、nport、ncos、host、hsrv、hport、hpri、hep、hque

##### 使用指南

无

##### 使用实例

# 查询glb模块下的CPB cell资源分配以及当前资源占用情况。

```
linux-4it5:~ # hinicadm pdm -i hnic0 -m glb
***** GLB RESOURCE *****
dyn_mode    max_sh    max_th      sh_cnt      total
0          21398     29488       0         19
```

#### 4.4.35.2 查询某个 index 下的 CPB Cell 资源分配以及当前资源占用情况 ( -m -x )

##### 命令功能

pdm -m -x命令用于查询某个index下的CPB cell资源分配以及当前资源占用情况。

##### 命令格式

**hinicadm pdm -i <devicename> -m <module name> -x <index>**

##### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hnic0、hnic1

参数	说明	取值
<i>module name</i>	指定显示资源类型。	all、glb、pg、tx、rx、yzone、rzone、nport、ncos、host、hsrv、hport、hpri、hep、hque
<i>index</i>	各类资源表项的索引，指定资源显示的表项索引。	-

## 使用指南

无

## 使用实例

```
# 查看host相关资源信息。  
[root@localhost ~]# hinicadm pdm -i hinic0 -m host -x 0  
***** HOST RESOURCE *****  
tbl_idx nm_th mqm_th gap_th nm_cnt  
host0 10292 10036 150 0
```

### 4.4.35.3 查询某段范围下的 CPB Cell 资源分配以及当前资源占用情况 ( -m -s -e )

## 命令功能

pdm -m -s -e命令用于查询某段范围下的CPB cell资源分配以及当前资源占用情况。

## 命令格式

```
hinicadm pdm -i <devicename> -m <module name> -s <start index> -e <end index>
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hinic0、hinic1
<i>module name</i>	指定显示资源类型。	all、glb、pg、tx、rx、yzone、rzone、nport、ncos、host、hsrv、hport、hpri、hep、hque
<i>start index</i>	各类资源显示的起始索引，每种类型有不同的index范围。	-
<i>end index</i>	各类资源显示的结束索引，每种类型有不同的index范围，要大于等于起始索引值。	-

## 使用指南

无

## 使用实例

```
# 查看某范围host相关资源信息。  
[root@localhost ~]# hinicadm pdm -i hinic0 -m host -s 0 -e 3  
***** HOST RESOURCE *****  
tbl_idx    nm_th      mqm_th     gap_th      nm_cnt  
host0      10292     10036       150        0  
host1      10292     10036       150        0  
host2      10292     10036       150        0  
host3      10292     10036       150        0
```

## 4.4.36 查询和设置 PFC 功能 ( pfc )

### 4.4.36.1 查询 PFC 功能

#### 命令功能

pfc命令用于查询设备的PFC功能。

#### 命令格式

```
hinicadm pfc -i <devicename>
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如: enp6s0、enp7s0

## 使用指南

无

## 使用实例

```
# 查询enp6s0的PFC是否开启。
```

```
[root@localhost ~]# hinicadm pfc -i enp6s0  
enp6s0      pfc state: disable  
pfcup:      1      1      1      1      1      1      1  
num TC's:   4
```

### 4.4.36.2 设置 PFC 功能 ( -e -p )

#### □ 说明

设置PFC功能之前，首先需要使能DCB功能，具体操作请参见[4.4.10 查询和设置DCB功能 \( dcb \)](#)。

## 命令功能

**pfc**命令用于设置设备的PFC功能。

## 命令格式

```
hinicadm pfc -i <devicename> -e <mode> [-p <priority>]
```

```
hinicadm pfc -i <devicename> -p <priority>
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如: enp6s0、enp7s0
<i>mode</i>	控制PFC功能的开关模式	<ul style="list-style-type: none"><li>• 1: 开启pfc</li><li>• 0: 关闭pfc</li></ul>
<i>priority</i>	设置0~7优先级PFC使能	xxxxxxxx <ul style="list-style-type: none"><li>• 1: 使能</li><li>• 0: 不使能</li></ul>

## 使用指南

当首次-e使能PFC后，再设置0~7优先级的PFC时，可以直接执行-p来实现。

## 使用实例

```
# 开启enp6s0的PFC功能。
```

```
[root@localhost ~]# hinicadm pfc -i enp6s0 -e 1
Set enp6s0 pfc to enable succeed
```

```
# 配置enp6s0的2、3优先级PFC使能。
```

```
[root@localhost ~]# hinicadm pfc -i enp6s0 -p 00110000
Set enp6s0 pfc to enable succeed.
Set enp6s0 pfc succeed
```

## 4.4.37 查询和设置用户优先级、cge 水线 ( qos )

## 命令功能

**qos**命令用于查询和设置指定设备的用户优先级、cge水线。

## 命令格式

```
hinicadm qos -i <devicename>
```

```
hinicadm qos -i <devicename> -c <qoscoss>
```

```
hinicadm qos -i <devicename> -t <cge pause time> -p <port id>
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询和设置的IN200在系统中的名称	例如：hinic0、hinic1
<i>qoscos</i>	待配置的用户优先级序列	<ul style="list-style-type: none"><li>• 序列中每个元素的值必须在0~7范围内</li><li>• 序列必须为递减模式</li><li>• 序列中包含的元素的个数必须与实际支持的cos数目一致（cos数目可通过查询命令获得）</li></ul>
<i>cge pause time</i>	当前芯片cge模块pause帧流控时间配置	-
<i>port id</i>	物理端口号	例如：0

## 使用指南

无

## 使用实例

# 设置IN200的用户优先级为6 5 3 0。

```
[root@localhost tool]# hinicadm qos -i hinic0 -c 6530
Set cos2up succeed.
```

# 查询IN200的用户优先级。

```
[root@localhost tool]# hinicadm qos -i hinic0
cos number: 4
user priority: 6 5 3 0
```

# 配置端口0的cge水线。

```
[root@localhost tool]# hinicadm qos -i hinic0 -t 65535 -p 0
qos command error(-6): Only cge card could set pause time.
```

## 4.4.38 查询和设置端口限速速率 ( rate )

### 命令功能

rate命令用于查询和设置指定IN200的指定端口的发送方向的限速参数。

### 命令格式

**hinicadm rate -i <devicename>**

**hinicadm rate -i <devicename> -l <limitvalue> [-d]**

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询/设置的IN200端口在系统中的网络名称	例如: eth0、eth1
<i>limitvalue</i>	端口发送方向限速速率与端口允许的最大速率的百分比	1~100的整数

## 使用指南

- 命令中带-d时，表示该设置为持久化配置，重启后依然有效。
- 命令中不带-d时，表示该设置非持久配置，在设备重启后失效，恢复为出厂配置。

## 使用实例

```
# 设置IN200的eth2的端口限速百分比为50%。
```

```
[root@localhost tool]# hinicadm rate -i eth2 -l 50
Set eth2 tx limit rate to 50% succeed
```

```
# 查询IN200的eth2的端口限速速率。
```

```
[root@localhost tool]# hinicadm rate -i eth2
eth2 tx limit rate: 50%.
eth2 default tx limit rate: 100%.
```

## 4.4.39 设置和触发网卡在 AC 掉电时待发送的报文数据 ( reboot\_notice )

### 命令功能

**reboot\_notice**命令用于设置或触发网卡在AC ( Alternating Current ) 掉电时待发送的报文数据。

### 命令格式

```
hinicadm reboot_notice -i <devicename> -m <mode> -f <filename>
hinicadm reboot_notice -i <devicename> -s
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如: hinic0、hinic1

参数	说明	取值
<i>mode</i>	读取或设置待发送报文数据	<ul style="list-style-type: none"><li>• <b>read</b>: 表示从卡端获取当前待发送报文数据，并存放在filename文件中。</li><li>• <b>write</b>: 表示向卡端设置待发送报文数据，数据内容从filename文件中读取。</li></ul>
<i>filename</i>	存放报文数据的文件名	-

## 使用指南

只有已打开的端口支持报文发送。

- **-s**表示触发一次网卡所有端口发送报文。
- 该命令只支持TaiShan 200服务器（型号 2280）适配的网卡，只支持linux系统。

## 使用实例

```
# 读取hinic0待发送的报文数据。  
[root@localhost ~]# hinicadm reboot_notice -i hinic0 -m read -f ./read.file  
File:read.file is created succeed  
reboot_notice command succeed.
```

```
# 向hinic0写入待发送的报文数据。  
[root@localhost ~]# hinicadm reboot_notice -i hinic0 -m write -f ./write.file  
reboot_notice command succeed.
```

```
# 触发一次网卡所有端口发送报文。  
[root@localhost ~]# hinicadm reboot_notice -i hinic0 -s  
reboot_notice command succeed.
```

### 4.4.40 恢复出厂设置 ( reset )

#### 命令功能

**reset**命令用于恢复IN200端口的出厂设置。

#### 命令格式

```
hinicadm reset -i <devicename> [-p <portid>]
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待恢复的IN200在系统中的名称	例如: hinic0、hinic1
<i>portid</i>	待恢复的端口号	-

## 使用指南

命令执行完毕后，需要重启OS使之生效。

## 使用实例

# 恢复IN200所有端口的出厂设置。

```
[root@localhost tool]#hinicadm reset -i hinic0
WARNING! reset operation will cause all data be cleared.
WARNING! You have selected to do reset operation.
Proceed with the reset? (Y|N): Y
Pf0 restore factory settings succeed.
Pf1 restore factory settings succeed.
Pf2 restore factory settings succeed.
Pf3 restore factory settings succeed.
Please reboot OS to take effect.
```

### 4.4.41 配置 ROCE\_DSCP 功能 ( roce\_dscp )

#### □ 说明

配置ROCE\_DSCP功能仅RoCE特性支持。

#### 4.4.41.1 查询 ROCE\_DSCP 功能

##### 命令功能

**roce\_dscp**命令用于查询设备的ROCE\_DSCP功能。

##### 命令格式

**hinicadm roce\_dscp -i <devicename>**

##### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如： hinic0、 hinic1

## 使用指南

无

## 使用实例

# 查询hinic0的ROCE\_DSCP是否开启。

```
[root@localhost ~]# hinicadm roce_dscp -i hinic0
Please set dscp enable.
```

#### 4.4.41.2 设置 ROCE\_DSCP 功能 ( -e )

##### 命令功能

**roce\_dscp -e**命令用于设置设备的ROCE\_DSCP功能。

##### 命令格式

**hinicadm roce\_dscp -i <devicename> -e <mode>**

##### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如: hinic0、hinic1
<i>mode</i>	控制ROCE_DSCP功能的开关模式	<ul style="list-style-type: none"><li>• 1: 开启ROCE_DSCP</li><li>• 0: 关闭ROCE_DSCP</li></ul>

##### 使用指南

无

##### 使用实例

# 开启hinic0的ROCE\_DSCP。

```
[root@localhost ~]# hinicadm roce_dscp -i hinic0 -e 1
Set hinic0 dscp enable succeed
hinic0 dscp state: enable
dscp2prio mapping:
  prio:0 dscp:0 1 2 3 4 5 6 7
  prio:1 dscp:8 9 10 11 12 13 14 15
  prio:2 dscp:16 17 18 19 20 21 22 23
  prio:3 dscp:24 25 26 27 28 29 30 31
  prio:4 dscp:32 33 34 35 36 37 38 39
  prio:5 dscp:40 41 42 43 44 45 46 47
  prio:6 dscp:48 49 50 51 52 53 54 55
  prio:7 dscp:56 57 58 59 60 61 62 63
```

#### 4.4.41.3 重置 ROCE\_DSCP 功能的优先级映射规则 ( -r )

##### 命令功能

**roce\_dscp -r**命令用于重置ROCE\_DSCP功能的优先级映射规则。

##### 命令格式

**hinicadm roce\_dscp -i <devicename> -r <mode>**

## 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如: hinic0、hinic1
<i>mode</i>	优先级映射规则	<ul style="list-style-type: none"><li>• 1: 重置ROCE_DSCP功能的优先级映射规则</li><li>• 0: 不重置ROCE_DSCP功能的优先级映射规则</li></ul>

## 使用指南

无

## 使用实例

# 重置ROCE\_DSCP功能的优先级映射规则。

```
[root@localhost ~]# hinicadm roce_dscp -i hinic0 -r 1
hinic0 dscp state: enable
Reset hinic0 dscp succeed
dscp2prio mapping:
  prio:0 dscp:0 1 2 3 4 5 6 7
  prio:1 dscp:8 9 10 11 12 13 14 15
  prio:2 dscp:16 17 18 19 20 21 22 23
  prio:3 dscp:24 25 26 27 28 29 30 31
  prio:4 dscp:32 33 34 35 36 37 38 39
  prio:5 dscp:40 41 42 43 44 45 46 47
  prio:6 dscp:48 49 50 51 52 53 54 55
  prio:7 dscp:56 57 58 59 60 61 62 63
```

### 4.4.41.4 设置 DSCP 与用户优先级的关系 ( -s -p )

#### 命令功能

**roce\_dscp -s -p**命令用于设置设备的DSCP值与优先级的映射关系。

#### 命令格式

```
hinicadm roce_dscp -i <devicename> -e <mode> -s <dscp> -p <priority>
hinicadm roce_dscp -i <devicename> -s <dscp> -p <priority>
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如: hinic0、hinic1

参数	说明	取值
<i>mode</i>	控制ROCE_DSCP功能的开关模式	<ul style="list-style-type: none"><li>• 1: 开启ROCE_DSCP</li><li>• 0: 关闭ROCE_DSCP</li></ul>
<i>dscp</i>	用户使用的dscp值	dscp取值0~63
<i>priority</i>	用户使用的优先级, 与 <a href="#">4.4.37 查询和设置用户优先级、cqe水线 ( qos ) 设置的优先级对应</a>	priority取值0~7

## 使用指南

无

## 使用实例

# 设置hinic0的dscp值26为优先级2的映射关系。

```
[root@localhost ~]# hinicadm roce_dscp -i hinic0 -s 26 -p 2
hinic0 dscp state: enable
dscp2prio mapping:
prio:0 dscp:0 1 2 3 4 5 6 7
prio:1 dscp:8 9 10 11 12 13 14 15
prio:2 dscp:16 17 18 19 20 21 22 23 26
prio:3 dscp:24 25 27 28 29 30 31
prio:4 dscp:32 33 34 35 36 37 38 39
prio:5 dscp:40 41 42 43 44 45 46 47
prio:6 dscp:48 49 50 51 52 53 54 55
prio:7 dscp:56 57 58 59 60 61 62 63
```

### 4.4.41.5 删除 DSCP 与用户优先级的映射关系 ( -d -p )

#### 命令功能

**roce\_dscp -d -p**命令用于删除设备的DSCP值与优先级的映射关系。

#### 命令格式

```
hinicadm roce_dscp -i <devicename> -e <mode> -d <dscp> -p <priority>
hinicadm roce_dscp -i <devicename> -d <dscp> -p <priority>
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如: hinic0、hinic0
<i>mode</i>	控制ROCE_DSCP功能的开关模式	<ul style="list-style-type: none"><li>• 1: 开启ROCE_DSCP</li><li>• 0: 关闭ROCE_DSCP</li></ul>

参数	说明	取值
<i>dscp</i>	用户使用的dscp值	dscp取值0~63
<i>priority</i>	用户使用的优先级，与 <a href="#">4.4.37 查询和设置用户优先级、cqe水线 ( qos ) 设置的优先级对应</a>	priority取值0~7

## 使用指南

无

## 使用实例

```
# 删除hinic0的dscp值26为优先级2的映射关系。
```

```
[root@localhost ~]# hinicadm roce_dscp -i hinic0 -d 26 -p 2
hinic0 dscp state: enable
dscp2prio mapping:
  prio:0 dscp:0 1 2 3 4 5 6 7
  prio:1 dscp:8 9 10 11 12 13 14 15
  prio:2 dscp:16 17 18 19 20 21 22 23
  prio:3 dscp:24 25 27 28 29 30 31
  prio:4 dscp:32 33 34 35 36 37 38 39
  prio:5 dscp:40 41 42 43 44 45 46 47
  prio:6 dscp:48 49 50 51 52 53 54 55
  prio:7 dscp:56 57 58 59 60 61 62 63
```

## 4.4.42 查询 RoCE 各端口发送接收实时流量 ( roce\_port\_traffic )

### 说明

查询RoCE各端口发送接收实时流量的功能仅RoCE特性支持。

## 命令功能

**roce\_port\_traffic**命令用于通过RoCE设备查询设备所在网卡各端口发送接收实时流量。

## 命令格式

```
hinicadm roce_port_traffic -i <device> -t <exec_time>
```

## 参数说明

参数	说明	取值
<i>device</i>	有效RoCE设备名称。	例如：hrn0_0、hrn0_bond_0
<i>exec_time</i>	执行时间。单位秒。	-

## 使用指南

无

## 使用实例

```
# 通过hrn0_bond_0实时显示所有RoCE端口流量状态，查看hrn0_bond_0所在网卡所有端口的流量数据。
```

```
[root@localhost ~]# hinicadm roce_port_traffic -i hrn0_bond_0 -t 2
| port tx(Mpps) tx(Gbps) rx(Mpps) rx(Gbps) | port tx(Mpps) tx(Gbps) rx(Mpps) rx(Gbps) |
| 0 0.000 0.000000 0.000 0.000000 | 1 0.000 0.000000 0.000 0.000000 |
| 2 0.004 0.000768 0.004 0.001024 | 3 0.000 0.000000 0.000 0.000000 |

| port tx(Mpps) tx(Gbps) rx(Mpps) rx(Gbps) | port tx(Mpps) tx(Gbps) rx(Mpps) rx(Gbps) |
| 0 0.000 0.000000 0.000 0.000000 | 1 0.000 0.000000 0.000 0.000000 |
| 2 0.004 0.003072 0.004 0.001024 | 3 0.000 0.000000 0.000 0.000000 |
```

## 4.4.43 查询和设置 SDI 配置 ( sdi\_cfg )

**sdi\_cfg**命令用于查询和设置SDI配置，包括管理vlan，PF限速参数。

### 4.4.43.1 查询配置的管理 VLAN ( -t cpath\_vlan )

#### 命令功能

**-t cpath\_vlan**命令用于查询配置的管理VLAN。

#### 命令格式

```
hinicadm sdi_cfg -i <devicename> -t cpath_vlan
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1

## 使用指南

无

## 使用实例

```
# 查询配置的管理VLAN。
```

```
linux-xvUVkf:~ # hinicadm sdi_cfg -i hinic0 -t cpath_vlan
None control path vlan configured.
```

### 4.4.43.2 配置管理 VLAN ( -t cpath\_vlan -v -s )

#### 命令功能

**-t cpath\_vlan-v -s**命令用于配置管理VLAN。

## 命令格式

```
hinicadm sdi_cfg -i <devicename> -t cpath_vlan -v <vlan> -s
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hinic0、hinic1
<i>vlan</i>	vlan号	整数形式：范围是0~4095

## 使用指南

无

## 使用实例

```
# 配置管理VLAN。  
linux-xvUVkf:~ # hinicadm sdi_cfg -i hinic0 -t cpath_vlan -v 4095 -s  
Set control path vlan 4095 succeed
```

### 4.4.43.3 删除配置的管理 VLAN ( -t cpath\_vlan -v -c )

## 命令功能

-t cpath\_vlan -v -c命令用于删除配置的管理VLAN。

## 命令格式

```
hinicadm sdi_cfg -i <devicename> -t cpath_vlan -v <vlan> -c
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hinic0、hinic1
<i>vlan</i>	vlan号	整数形式：范围是0~4095

## 使用指南

无

## 使用实例

```
# 删除配置的管理VLAN。  
  
linux-xvUVkf:~ # hinicadm sdi_cfg -i hinic0 -t cpath_vlan -v 4095 -c  
Clear control path vlan 4095 succeed.
```

### 4.4.43.4 查询指定 PF 发送方向限速参数 ( -t pf\_tx\_rate )

#### 命令功能

-t pf\_tx\_rate 命令用于查询指定PF发送方向限速参数。

#### 命令格式

```
hinicadm sdi_cfg -i <devicename> -t pf_tx_rate
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1

## 使用指南

无

#### 使用实例

```
# 查询指定设备eth4发送方向限速参数。  
  
linux-xvUVkf:~ # hinicadm sdi_cfg -i eth4 -t pf_tx_rate  
eth4 pf_tx_rate 100000000kbps,2560000kbps,100000000kbps,2560000kbps.
```

### 4.4.43.5 设置指定 PF 发送方向限速参数 ( -t pf\_tx\_rate -l )

#### 命令功能

-t pf\_tx\_rate -l 命令用于设置指定PF发送方向限速参数。

#### 命令格式

```
hinicadm sdi_cfg -i <devicename> -t pf_tx_rate -l <cir,cbs,pir,pbs>
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hinic0、hinic1

参数	说明	取值
<i>cir,cbs,pir,pbs</i>	tx限速参数	<ul style="list-style-type: none"><li>• cir、pir的单位是kbps。</li><li>• cbs、pbs的单位是kbit。</li></ul>

## 使用指南

无

## 使用实例

# 设置指定设备eth4发送方向限速参数。

```
linux-xvUVkf:~ # hinicadm sdi_cfg -i eth4 -t pf_tx_rate -l 1000000,800000,1000000,800000
Set eth4 pf rate succeed.
```

### 4.4.43.6 删除指定 PF 发送方向限速参数 ( -t pf\_tx\_rate -c )

#### 命令功能

-t pf\_tx\_rate -c命令用于删除指定PF发送方向限速参数。

#### 命令格式

```
hinicadm sdi_cfg -i <devicename> -t pf_tx_rate -c
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hinic0、hinic1

## 使用指南

无

## 使用实例

# 删除指定设备eth4发送方向限速参数。

```
linux-xvUVkf:~ # hinicadm sdi_cfg -i eth4 -t pf_tx_rate -c
Clear eth4 pf rate limit succeed.
```

### 4.4.43.7 查询指定 PF 接收方向限速参数 ( -t pf\_rx\_rate )

#### 命令功能

-t pf\_rx\_rate命令用于查询指定PF接收方向限速参数。

## 命令格式

```
hinicadm sdi_cfg -i <devicename> -t pf_rx_rate
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1

## 使用指南

无

## 使用实例

```
# 查询指定设备eth4接收方向限速参数。
```

```
linux-xvUVkf:~ # hinicadm sdi_cfg -i eth4 -t pf_rx_rate
eth4 pf_rx_rate 100000000kbps,2560000kbits.
```

## 4.4.43.8 设置指定 PF 接收方向限速参数 ( -t pf\_rx\_rate -l )

### 命令功能

-t pf\_rx\_rate -l命令用于设置指定PF接收方向限速参数。

### 命令格式

```
hinicadm sdi_cfg -i <devicename> -t pf_rx_rate -l <cir,cbs>
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hinic0、hinic1
<i>cir,cbs</i>	tx限速参数。 • cir的单位是kbps • cbs的单位是kbit	

## 使用指南

无

## 使用实例

```
# 设置指定设备eth4接收方向限速参数。
```

```
linux-xvUVkf:~ # hinicadm sdi_cfg -i eth4 -t pf_rx_rate -l 1000000,800000
Set eth4 pf rate succeed.
```

#### 4.4.43.9 删除指定 PF 接收方向限速参数 ( -t pf\_rx\_rate -c )

##### 命令功能

-t pf\_rx\_rate -c命令用于删除指定PF接收方向限速参数。

##### 命令格式

```
hinicadm sdi_cfg -i <devicename> -t pf_rx_rate -c
```

##### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hinic0、hinic1

##### 使用指南

无

##### 使用实例

```
# 删除指定设备eth4接收方向限速参数。
```

```
linux-xvUVkf:~ # hinicadm sdi_cfg -i eth4 -t pf_rx_rate -c
Clear eth4 pf rate limit succeed.
```

#### 4.4.43.10 查询 SDI 卡服务器侧重启，芯片是否跟随复位 ( -t reset\_scheme )

##### 命令功能

-t reset\_scheme命令用于查询SDI卡服务器侧重启，芯片是否跟随复位。

##### 命令格式

```
hinicadm sdi_cfg -i <devicename> -t reset_scheme
```

##### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1

## 使用指南

无

## 使用实例

```
# 查询SDI卡服务器侧重启，芯片是否跟随复位。
```

```
linux-exSeNf:~ # hinicadm sdi_cfg -i hinic1 -t reset_scheme
Current reset scheme: enable
```

### 4.4.43.11 设置 SDI 卡服务器侧重启，芯片是否跟随复位（-t reset\_scheme -e）

#### 命令功能

-t reset\_scheme -e命令用于设置SDI卡服务器侧重启，芯片是否跟随复位。

#### 命令格式

```
hinicadm sdi_cfg -i <devicename> -t reset_scheme -e <enable: 1/disable: 0>
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hinic0、hinic1
<i>enable: 1/disable: 0</i>	芯片是否跟随复位。	<ul style="list-style-type: none"><li>• 1: 复位</li><li>• 0: 不复位。</li></ul>

## 使用指南

- 该设置只对mult-host卡生效。
- 对普通的single-host卡可以设置，但是并不会生效，芯片始终保持联动复位的状态。

## 使用实例

```
# 设置SDI卡服务器侧重启，芯片跟随复位。
```

```
linux-exSeNf:~ # hinicadm sdi_cfg -i hinic1 -t reset_scheme -e 0
Set reset scheme disable succeed.
```

### 4.4.43.12 查询 SDI 卡上服务器侧 PT 设备的显示状态（-t set\_pt）

#### 命令功能

-t set\_pt命令用于查询SDI卡服务器侧PT（PassThrough）设备的显示状态。

## 命令格式

```
hinicadm sdi_cfg -i <devicename> -t set_pt
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1

## 使用指南

无

## 使用实例

```
# 查询SDI卡服务器侧PT设备的显示状态，实例中为隐藏。
```

```
linux-SUKrdK:~ # hinicadm sdi_cfg -i hinic1 -t set_pt
Current pt status: hide
```

### 4.4.43.13 设置 SDI 卡服务器侧 PT 设备的显示状态（-t set\_pt -d）

## 命令功能

-t set\_pt -d命令用于设置SDI卡服务器侧PT设备的显示状态。

## 命令格式

```
hinicadm sdi_cfg -i <devicename> -t set_pt -d <display:1 hide:0>
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hinic0、hinic1
<i>display:1 hide:0</i>	服务器侧PT设备的显示状态。 ● 1：显示 ● 0：隐藏	

## 使用指南

无

## 使用实例

```
# 设置SDI卡服务器侧PT设备为显示模式。
```

```
linux-SUKrdK:~ # hinicadm sdi_cfg -i hinic1 -t set_pt -d 1
Display pt status succeed.
```

## 4.4.44 查询和设置 SDI 卡模式 ( sdi\_mode )

### 4.4.44.1 查询 SDI 卡模式

#### 命令功能

**sdi\_mode**命令用于查询SDI卡模式。

#### 命令格式

```
hinicadm sdi_mode -i <devicename>
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如: hinic0、hinic1

#### 使用指南

无

#### 使用实例

# 查询SDI卡的模式。

```
linux-EXVCVE:~ # hinicadm sdi_mode -i hinic0
Current sdi mode: vm
SDI mode in permanent configuration: vm
```

### 4.4.44.2 设置 SDI 卡模式 ( -m )

#### 命令功能

**sdi\_mode -m**命令用于查询和设置SDI卡模式。

#### 命令格式

```
hinicadm sdi_mode -i <devicename> -m <mode>
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如: hinic0、hinic1

参数	说明	取值
<i>mode</i>	SDI卡的模式。	<ul style="list-style-type: none"><li>• vm: 虚拟机模式</li><li>• bm: 裸机模式</li></ul>

## 使用指南

无

## 使用实例

```
# 设置SDI卡的模式。  
linux-EXvCVE:~ # hinicadm sdi_mode -i hinic0 -m bm  
Set sdi mode to bm succeed.  
Please reboot OS for the new mode to take effect.
```

## 4.4.45 查询和设置端口协商自适应状态 ( self\_adaption )

### 4.4.45.1 查询端口协商自适应状态 ( -p )

#### 命令功能

**self\_adaption -p**命令用于查询指定端口的协商自适应模式的启用状态。

#### 命令格式

```
hinicadm self_adaption -i <devicename> -p <portid>
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如: hinic0、hinic1
<i>portid</i>	待查询的端口号	-

## 使用指南

恢复为自适应的出厂设置。

## 使用实例

```
# 查询port0的协商自适应模式的开启状态。
```

```
[root@localhost tool]# hinicadm self_adaption -i hinic0 -p 0  
Port 0 link mode adaptation : on
```

### 4.4.45.2 清除配置 ( -p -c )

#### 命令功能

**self\_adaption -p -c**命令用于清除配置，使其恢复出厂设置。

#### 命令格式

```
hinicadm self_adaption -i <devicename> -p <portid> [-c]
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hnic0、hnic1
<i>portid</i>	待设置的端口号	-

#### 使用指南

无

#### 使用实例

# 清除port0的配置使其设置为默认。

```
linux-4it5:~ # hinicadm self_adaption -i hnic0 -p 0 -c
Clear default configuration succeed.
Active default configuration succeed.
Clear hilink default configuration succeed.
```

### 4.4.45.3 设置端口协商自适应状态 ( -p -s -t -d )

#### 命令功能

**self\_adaption -p -s**命令用于开关自适应的启用状态。

**self\_adaption -p -s -t**命令用于指定自适应的速率类型。

#### 命令格式

```
hinicadm self_adaption -i <devicename> -p <portid> -s <status> [-t <type>]
hinicadm self_adaption -i <devicename> -p <portid> -s <status> [-t <type>] [-d]
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待设置的IN200在系统中的名称	例如：hnic0、hnic1

参数	说明	取值
<i>portid</i>	待设置的端口号	-
<i>status</i>	协商自适应模式的开启状态	<ul style="list-style-type: none"><li>• on: 开启</li><li>• off: 关闭 (默认)</li></ul>
<i>type</i>	指定自适应的速率类型	<ul style="list-style-type: none"><li>• low</li><li>• high</li></ul>

## 使用指南

- 命令中带-d时，表示该设置为持久化配置，重启后依然有效。
- 命令中不带-d时，表示该设置非持久配置，在设备重启后失效，恢复为出厂配置。

## 使用实例

# 指定自适应的速率类型。

```
linux-4it5:~ # hinicadm self_adaption -i hinic0 -p 0 -s on -t high
Set port 0 link mode adaptation to on succeed.
```

# 开启port0的协商自适应模式并持久生效。

```
[root@localhost tool]# hinicadm self_adaption -i hinic0 -p 0 -s on -d
Set default link mode adaptation succeed.
Active default configuration succeed.
Set default link cfg succeed.
```

## 4.4.46 查询指定配置类型的 serdes 信息 ( serdes )

### 命令功能

**serdes**命令用于查询指定配置类型的serdes信息。

### 命令格式

```
hinicadm serdes -i <devicename>
```

```
hinicadm serdes -i <devicename> -t <hilink_type> [-m <macro>]
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如: hinic0、eth0、eth1
<i>hilink_type</i>	配置类型	<ul style="list-style-type: none"><li>• 0: hi16和hi30</li><li>• 1: hi16或hi30</li></ul>
<i>macro</i>	serdes的子模块	0~5

## 使用指南

当输入参数为IN200的名称时，获取到的是IN200上所有端口的serdes信息。

## 使用实例

# 查询所有serdes信息。

```
linux-4it5:~ # hinicadm serdes -i hinic0
-----show hi16 info-----
macro is 0 dsnum is 0
cs0_csr2 = 0 (success)
cs0_csr_51 = 0 (success)
cs0_csr54 = 13422
cs0_csr63 = 584
cs1_csr2 = 0 (success)
.....
```

# 查询hinic0的serdes信息。

```
[root@localhost tool]# hinicadm serdes -i hinic0 -t 0
-----show hi16 info-----
macro is 0 dsnum is 0
cs0_csr2 = 0 (success)
cs0_csr_51 = 0 (success)
cs0_csr54 = 11882
cs0_csr63 = 0
cs1_csr2 = 0 (success)
cs1_csr_51 = 0 (success)
cs1_csr54 = 6254
cs1_csr63 = 529
dsclk_csr0 = 0 (success)
dsclk_csr24 = 0 (success)
dsclk_csr27 = 38352
tx_csr2 = 180
tx_csr26 = 51100
rx_csr10 = 180
rx_csr26 = 24703
rx_csr61 = 6164
tx_csr_rw_result = 0 (success)
rx_csr_rw_result = 0 (success)
tx_csr48 = 49163
rx_csr10 = 1
eye_top = 15
.....
```

# 查询hinic0的serdes对应macro的信息。

```
linux-4it5:~ # hinicadm serdes -i hinic0 -t 1 -m 0
-----show hi16 info-----
macro is 0 dsnum is 0
cs0_csr2 = 0 (success)
cs0_csr_51 = 0 (success)
cs0_csr54 = 13422
cs0_csr63 = 583
```

### 4.4.47 查询指定端口的光模块信息 ( sfp )

#### 命令功能

**sfp**命令用于查询指定端口上安装的光模块信息。

## 命令格式

```
hinicadm sfp -i <devicename> -p <portid> [-a]
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1
<i>portid</i>	待查询的端口号	-

## 使用指南

带-a参数时，表示打印光模块所有信息，回显信息展现为十六进制数据。

## 使用实例

```
# 查询port0的光模块信息。
```

```
[root@localhost tool]#hinicadm sfp -i hinic0 -p 0
-----show sfp info-----
ucId = 0x03
ucIdExt = 0x04
ucConnector = 0x21
ucEncoding = 6
ucBrNominal = 255 (100MBd)
ucRateIdentifier = 0
ucLengthSmfKm = 0(km)
ucLengthSmf = 0(100m)
ucLengthSmfOm2 = 0(10m)
ucLengthSmfOm1 = 0(10m)
ucLengthCable = 3(m)
ucLengthOm3 = 0(m)
aucVendorName = Amphenol
aucVendorSN = 0215761612073874
ucTransceiver = 0
aucVendorOui = 78 a7 14
aucVendorPn = NDCCGF-H203
aucVendorRev = A
aucWaveLength = 256
ucUnAllocated = 0
ucCcBase = 35
aucTemperature = NA (DAC)
```

## 4.4.48 查询和设置指定端口的虚拟化控制模式的启用状态（sriov）

### 命令功能

sriov命令用于查询和设置指定端口的虚拟化控制模式。

## 命令格式

```
hinicadm sriov -i <devicename> -p <portid>
hinicadm sriov -i <devicename> -p <portid> -s <sriov>
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询/设置的IN200在系统中的名称	例如: hinic0、hinic1
<i>portid</i>	待查询/设置的端口号	-
<i>sriov</i>	虚拟化控制模式的开启状态	<ul style="list-style-type: none"><li>● enable</li><li>● disable</li></ul>

## 使用指南

无

## 使用实例

# 查询port0的虚拟化控制模式的开启状态。

```
[root@localhost tool]# hinicadm sriov -i hinic0 -p 0
Pf0 sriov control status: disable.
```

# 开启port0的虚拟化控制模式。

```
[root@localhost tool]# hinicadm sriov -i hinic0 -p 0 -s enable
Set pf0 sriov control to enable succeed.
```

## 4.4.49 查询指定设备的线性表 ( table )

### 命令功能

table命令用于查询指定设备的线性表信息。

### 命令格式

```
hinicadm table -i <devicename> -t <tabletype>
```

```
hinicadm table -i <devicename> -t <tabletype> -x <tableindex>
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如: hinic0、hinic1

参数	说明	取值
<i>tabletype</i>	待查询的线性表类型	<ul style="list-style-type: none"><li>• 0: MAC表</li><li>• 1: 全局表</li><li>• 2: function配置表</li><li>• 3: port配置表</li><li>• 4: VLAN表</li><li>• 5: FDIR_CTRL表</li><li>• 7: FDIR_IO表</li><li>• 9: FDIR_TCAM表</li><li>• 10: RSS_CONTEXT表</li><li>• 11: ELB表</li></ul>
<i>tableindex</i>	线性表索引号	<ul style="list-style-type: none"><li>• <i>tabletype</i>为1时取0-63</li><li>• <i>tabletype</i>为2时取0-511</li><li>• <i>tabletype</i>为3时取0-7</li><li>• <i>tabletype</i>为4时取0-4095</li></ul>

## 使用指南

无

## 使用实例

# 查询IN200的线性表信息。

```
[root@localhost tool]# hinicadm table -i hinic0 -t 0
static er_id mac          vlan_id forward_type forward_id
1   1   xx.xx.xx.xx.xx.xx  0     6      9
1   1   xx.xx.xx.xx.xx.xx  0     6      8
1   3   xx.xx.xx.xx.xx.xx  0     0      3
1   2   xx.xx.xx.xx.xx.xx  0     0      2
1   1   xx.xx.xx.xx.xx.xx  0     0      1
1   0   xx.xx.xx.xx.xx.xx  0     0      0
```

# 查询IN200指定类型的某个线性表信息。

```
[root@localhost sbin]# hinicadm table -i hinic0 -t 4 -x 12
node: 10 instance : 3 entry_size : 16 index : 12.
tag_sml_vlan_tbl_elem[0].learn_en : 0
tag_sml_vlan_tbl_elem[0].elb_index : 0
tag_sml_vlan_tbl_elem[1].learn_en : 0
tag_sml_vlan_tbl_elem[1].elb_index : 0
tag_sml_vlan_tbl_elem[2].learn_en : 0
tag_sml_vlan_tbl_elem[2].elb_index : 0
tag_sml_vlan_tbl_elem[3].learn_en : 0
tag_sml_vlan_tbl_elem[3].elb_index : 0
tag_sml_vlan_tbl_elem[4].learn_en : 0
tag_sml_vlan_tbl_elem[4].elb_index : 0
tag_sml_vlan_tbl_elem[5].learn_en : 0
```

```
tag_sml_vlan_tbl_elem[5]_elb_index : 0
tag_sml_vlan_tbl_elem[6]_learn_en : 0
tag_sml_vlan_tbl_elem[6]_elb_index : 0
tag_sml_vlan_tbl_elem[7]_learn_en : 0
tag_sml_vlan_tbl_elem[7]_elb_index : 0
```

## 4.4.50 查询芯片温度和光模块温度 ( temperature )

### 命令功能

**temperature**命令用于查询指定IN200的芯片温度和光模块的温度。

### 命令格式

```
hinicadm temperature -i <devicename>
```

### 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1

### 使用指南

无

### 使用实例

# 查询IN200的芯片温度和光模块温度。

```
[root@localhost tool]# hinicadm temperature -i hinic0
current temperature (unit: degree centigrade)
controller: 55
sfp1: 59
sfp2: absent
sfp3: NA
sfp4: 56
```

## 4.4.51 查询微码线程状态 ( tile\_io )

### 命令功能

**tile\_io**命令用于查询IN200的微码线程状态。

### 命令格式

```
hinicadm tile_io -i <devicename>
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如：hinic0、hinic1

## 使用指南

无

## 使用实例

# 查询IN200的微码线程状态。

```
[root@localhost tool]#hinicadm tile_io -i hinic0
***** tile IO status *****
core0_tc0_IO_0 : WAIT core0_tc1_IO_0 : WAIT core0_tc2_IO_0 : WAIT core0_tc3_IO_0 : WAIT
core1_tc0_IO_0 : WAIT core1_tc1_IO_0 : WAIT core1_tc2_IO_0 : WAIT core1_tc3_IO_0 : WAIT
core2_tc0_IO_0 : WAIT core2_tc1_IO_0 : WAIT core2_tc2_IO_0 : WAIT core2_tc3_IO_0 : WAIT
core3_tc0_IO_0 : WAIT core3_tc1_IO_0 : WAIT core3_tc2_IO_0 : WAIT core3_tc3_IO_0 : WAIT
core4_tc0_IO_0 : WAIT core4_tc1_IO_0 : WAIT core4_tc2_IO_0 : WAIT core4_tc3_IO_0 : WAIT
core5_tc0_IO_0 : WAIT core5_tc1_IO_0 : WAIT core5_tc2_IO_0 : WAIT core5_tc3_IO_0 : WAIT
core6_tc0_IO_0 : WAIT core6_tc1_IO_0 : WAIT core6_tc2_IO_0 : WAIT core6_tc3_IO_0 : WAIT
core7_tc0_IO_0 : WAIT core7_tc1_IO_0 : WAIT core7_tc2_IO_0 : WAIT core7_tc3_IO_0 : WAIT
core8_tc0_IO_0 : WAIT core8_tc1_IO_0 : WAIT core8_tc2_IO_0 : WAIT core8_tc3_IO_0 : WAIT
```

## 4.4.52 升级固件 ( updatefw )

### 4.4.52.1 冷升级整体固件 ( -f )

#### 命令功能

**updatefw -f**命令用于冷升级整体固件包，升级完成后需要重启OS使之生效。

#### 命令格式

```
hinicadm updatefw -i <devicename> -f <fwfile>
```

#### 参数说明

参数	说明	取值
<i>devicename</i>	待升级的IN200在系统中的名称	例如：hinic0、hinic1
<i>fwfile</i>	固件升级包路径	格式：“路径/升级包名称”

## 使用指南

无

## 使用实例

# 冷升级方式升级IN200固件。

```
[root@localhost tool]#hinicadm updatefw -i hinic0 -f Hi1822_nic_prd_1h_4x25G.bin
Please do not remove driver or network device.
Loading...
Firmware update start: 2018-07-14 22:22:48
[>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>] [100%][\]
Firmware update finish: 2018-07-14 22:23:10
Firmware update time used: 22s
Loading firmware image succeed.
Please reboot OS to take firmware effect
```

### 4.4.52.2 冷升级 FW 固件及配置文件 ( -f -c )

#### 命令功能

**updatefw -f -c**命令用于冷升级FW固件及配置文件。

##### 说明

当前适配IN200的固件包为整体升级包，因此对其升级时请勿使用此命令，请执行[冷升级整体固件 \(-f\)](#)。

#### 命令格式

**hinicadm updatefw -i <devicename> -f <fwfile> -c <configfile>**

#### 参数说明

参数	说明	取值
<i>devicename</i>	待升级的IN200在系统中的名称	例如： hinic0、 hinic1
<i>fwfile</i>	Firmware固件升级包路径	格式：“路径/升级包名称”
<i>configfile</i>	配置文件升级包路径	格式：“路径/升级包名称”

## 使用指南

无

## 使用实例

# 冷升级方式升级IN200的FW固件及配置文件。

```
[root@localhost tool]#hinicadm updatefw -i hinic0 -f /OSM/modules/port/Hi1822_fw.bin -c /OSM/
modules/Hi1822/update_cfg/Hi1822_cfg_nic_smart_4x25G.bin
```

#### 4.4.52.3 热升级固件 (-f -a)

## 命令功能

**updatefw -f -a**命令仅支持热补丁固件包的热升级，并选择激活方式。热补丁固件包的详细说明请参考对应版本说明书。

命令格式

**hinicadm updatefw -i <devicename> -f <fwfile> -a <activemode> [-t<activitytype>]**

## 参数说明

参数	说明	取值
<i>devicename</i>	待升级的IN200在系统中的名称	例如： hinic0、hinic1
<i>fwfile</i>	固件升级包路径	格式：“路径/升级包名称”
<i>activemode</i>	激活模式	<ul style="list-style-type: none"><li>• now：立即激活</li><li>• later：延迟激活</li></ul> <p><b>说明</b></p> <ul style="list-style-type: none"><li>• 若未输入-a参数，则命令功能等同于<b>冷升级整体固件</b>。</li><li>• 选择延迟激活后，请用户在适当的时间重新执行“立即激活”。</li></ul>
<i>activetype</i>	激活类型	<ul style="list-style-type: none"><li>• up：仅激活uP ( micro Processor ) 固件</li><li>• ucode：仅激活微码 ( micro Code ) 固件</li></ul> <p><b>说明</b></p> <p>若未输入-t参数，则表示全部激活。</p>

使用指南

无

使用实例

# 升级IN200固件并激活。

```
[root@localhost tool]# hinicadm updatefw -i hinic0 -f ./Hi1822_nic_std_4_25G.bin -a now  
Please do not remove driver or network device.
```

```
Loading...
Firmware update start: 2018-07-14 22:54:47
[>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>] [100%][\]
Firmware update finish: 2018-07-14 22:55:07
Firmware update time used: 20s
Do not operate the device during the hot upgrade.
Firmware is activating. Please wait...  

```

#### 4.4.52.4 查询当前 IN200 的固件能否被指定版本热升级 ( -v -t )

##### 命令功能

**updatefw -a**命令用于查询当前IN200的固件能否被指定版本热升级。

##### 命令格式

```
hinicadm updatefw -i <devicename> -v <target version> -t<activetype>
```

##### 参数说明

参数	说明	取值
<i>target version</i>	指定版本	-
<i>activetype</i>	激活类型	<ul style="list-style-type: none"><li>• up: 仅激活uP固件</li><li>• ucode: 仅激活微码固件</li></ul>

##### 使用指南

无

##### 使用实例

#查询当前IN200的固件能否被指定版本号的版本热升级，本例中3.2.0.5的up版本可支持当前硬件的热升级。

```
linux-4it5:~ # hinicadm updatefw -i hinic0 -v 3.2.0.5 -t up
Version check succeed.
The running version can be hot upgraded to the target version.
```

# 查询当前IN200的固件能否被指定版本号的版本热升级，本例中3.2.0.5的up版本不支持当前硬件的热升级。

```
linux-4it5:~ # hinicadm updatefw -i hinic0 -v 2.3.2.13 -t up
UP can't be hot upgraded.
updatefw command error(-6): The running version and the target version match failed.
```

#### 4.4.53 查询设备版本信息 ( version )

##### 命令功能

**version**命令用于查询IN200驱动、固件、工具的版本信息。

## 命令格式

**hinicadm version -i <devicename>**

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如： hinic0、 hinic1

## 使用指南

回显中主要字段的含义如表4-2所示。

表 4-2 回显说明

命令格式	字段	含义	举例
<b>hinicadm version -i &lt;devicename&gt;</b>	boot	boot版本信息。	2.4.1.0 2019-11-15_21:42:19
	up	网卡固件版本信息（MPU）。	2.6.0.10 2020-02-18_17:34:19
	ucode	网卡固件版本信息（微码）。	2.6.0.10 2020-02-18_17:37:31
	hinicadm tool	hinicadm工具版本信息。	2.8.0.3 2020-03-06_21:00:04
	hinic driver	网卡驱动版本信息。	2.8.0.3 2020-03-09_10:04:21

## 使用实例

# 查询指定IN200的版本信息。

```
[root@localhost tool]# hinicadm version -i hinic0
boot: 2.4.1.0 2019-11-15_21:42:19
up: 2.6.0.10 2020-02-18_17:34:19
ucode: 2.6.0.10 2020-02-18_17:37:31
hinicadm tool: 2.8.0.3 2020-03-06_21:00:04
hinic driver: 2.8.0.3 2020-03-09_10:04:21
```

## 4.4.54 管理 SFP/QSFP 的信息和状态 ( xsfp )

### 命令功能

**xsfp**命令用于管理SFP/QSFP信息和状态。

## 命令格式

```
hinicadm xsfp -i <devicename> [-f <function>] [-p <port id>] [-s <status>]
```

## 参数说明

参数	说明	取值
<i>devicename</i>	待查询的IN200在系统中的名称	例如： hinic0、 hinic1
<i>function</i>	查询或设置SFP/QSFP的功能类型	<ul style="list-style-type: none"><li>• 0：查询SFP/QSFP的信息</li><li>• 1：设置SFP/QSFP的使能状态</li></ul>
<i>port id</i>	物理端口号	例如： 0
<i>status</i>	设置SFP/QSFP的使能状态	<ul style="list-style-type: none"><li>• 0：不使能SFP/QSFP的使能状态</li><li>• 1：使能SFP/QSFP的使能状态</li></ul>

## 使用指南

无。

## 使用实例

```
# 开启hinic0的端口0的SFP/QSFP光模块的使能状态。
```

```
root@ubuntu:~# hinicadm xsfp -i hinic0 -f 1 -p 0 -s 1
Set optical module status: enable succeed.
```

```
# 关闭hinic0的端口0的SFP/QSFP光模块的使能状态。
```

```
root@ubuntu:~# hinicadm xsfp -i hinic0 -f 1 -p 0 -s 0
Set optical module status: disable succeed.
```

```
# 查看hinic0的端口0的SFP/QSFP光模块信息。
```

```
root@ubuntu:~# hinicadm xsfp -i hinic0 -f 0 -p 0
Original eeprom data:
0x3 0x4 0x21 0x1 0x0 0x0 0x0 0x41 0x4 0x80
0x0 0x6 0xff 0x0 0x0 0x0 0x0 0x0 0x1 0x0
0x4c 0x55 0x58 0x53 0x48 0x41 0x52 0x45 0x2d 0x49
0x43 0x54 0x20 0x20 0x20 0xd 0x3c 0x18 0xa0
0x4c 0x39 0x39 0x53 0x46 0x30 0x32 0x34 0x2d 0x53
0x44 0x2d 0x52 0x20 0x20 0x20 0x42 0x20 0x20 0x20
0x1 0x0 0x0 0x21 0x0 0x0 0x67 0x0 0x5a 0x30
0x30 0x41 0x32 0x52 0x31 0x37 0x35 0x30 0x30 0x30
0x30 0x30 0x30 0x36 0x31 0x37 0x31 0x32 0x31 0x31
0x20 0x20 0x0 0x0 0x8 0x4e 0x0 0x0 0x0 0x0
0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
...
...
0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
```

```
Vendor: LUXSHARE-ICT
Vendor PN: L99SF024-SD-R
Vendor Date: 171211
Serial Num: Z00A2R1750000006
Present: 1
Status: 0xff

Temp high thres: -1
Temp low thres: -1
Temperature: -1
Channel Num: 0
TxPower High Thres: 4294967295
TxPower Low Thres: 4294967295
RxPower High Thres: 4294967295
RxPower Low Thres: 4294967295
Vcc High Thres: 4294967295
Vcc Low Thres: 4294967295
Vcc: 4294967295
TxBias High Thres: 4294967295
TxBias Low Thres: 4294967295

Sfp Mode: 4294967295
Speed: 25750
CableType: 2
Speed Map: 0x0
bom:
model:
rev:
```

## 4.5 升级 hinicadm

### 前提条件

- 已下载hinicadm工具的安装包。  
安装包的名称和获取方法可参见[3.1 获取软件包](#)。
- 安装包已上传到服务器OS。

### Linux 系统下升级 hinicadm

**步骤1** 登录服务器OS。

**步骤2** 升级hinicadm工具。

- rpm包的升级：

执行**rpm -Uvh hinicadm-xxxx-xxxx.xxx.rpm**。

例如：

```
linux-ueeJIR:/Hi1822/tools/linux # rpm -Uvh hinicadm-1.2.8.0-1.x86_64.rpm
Preparing... ################################################ [100%]
Updating / installing...
1:hinicadm-1.2.8.0-1 ################################################ [ 50%]
Cleaning up / removing...
2:hinicadm-1.2.7.0-1 ################################################ [100%]
```

- deb包的升级

执行**dpkg -i hinicadm-xxxx-xxxx.xxx.deb**。

例如：

```
root@ubuntu1804:/home/ubuntu # dpkg -i hinicadm-1.6.1.2-1.aarch64.deb
(Reading database ... 69641 files and directories currently installed.)
Preparing to unpack hinicadm-1.6.1.2-1.aarch64.deb ...
Unpacking hinicadm (1.6.1.2) over (1.6.1.1) ...
Uninstall hinicadm tool successfully.
```

```
NOTE: There are some user files under the directory /opt/hinic/.  
If you don't need them, you can delete them manually.  
Setting up hinicadm (1.6.1.2) ...  
Installing... Please wait for a moment.  
Install hinicadm tool successfully.
```

----结束

## 后续操作

升级完成后，您可以通过**hinicadm -v**命令查询工具的当前版本，以保证升级正确。

## 4.6 卸载 hinicadm

### Linux 系统下卸载 hinicadm

**步骤1** 登录服务器OS。

**步骤2** 进入工具安装路径。

**步骤3** 卸载hinicadm工具。

- rpm包的卸载：

执行**rpm -e hinicadm**。

```
linux-ueFIIR:/Hi1822/tools/linux # rpm -e hinicadm  
NOTE: There are some user files under the directory /opt/hinic/.  
If you don't need them, you can delete them manually.
```

- deb包的卸载：

执行**dpkg -r hinicadm**。

```
root@ubuntu1804:/home/ubuntu# dpkg -r hinicadm  
(Reading database ... 69641 files and directories currently installed.)  
Removing hinicadm (1.6.1.2) ...  
Uninstall hinicadm tool successfully.  
NOTE: There are some user files under the directory /opt/hinic/.  
If you don't need them, you can delete them manually.
```

----结束

# 5 标准管理工具

ethtool、ifconfig、ip link、vconfig、dcbtool等工具为linux系统自带的网卡管理工具，IN200作为标准的PCIe卡，用户可以使用ethtool等工具对其进行基本的管理配置。

IN200支持的常用ethtool命令及其功能如表5-1所示。关于ethtool指定命令的详细使用方法，可通过命令行自带的帮助信息获取。

表 5-1 常用 ethtool 命令

命令	功能
ethtool ethx	显示网口基本设置。
ethtool -s ethx	设置网口自协商模式及网口速率。 <b>说明</b> 使用DAC线缆连接IN200端口时，若对端设备关闭自协商，且强制速率为25GE，需要对IN200的对应端口执行 <b>ethtool -s ethx autoneg off speed 25000</b> 命令关闭自协商，同时设置强制速率为25GE。
ethtool -a ethx	显示网口pause状态。
ethtool -A ethx	设置网口pause状态。
ethtool -c ethx	显示中断聚合参数。
ethtool -C ethx	设置中断聚合参数。
ethtool -g ethx	显示队列深度信息。
ethtool -G ethx	设置队列深度信息。
ethtool -k ethx	显示网卡支持的卸载特性和加速特性。

命令	功能
ethtool -K ethx	设置网卡支持的卸载特性和加速特性。
ethtool -i ethx	显示网口相关信息。
ethtool -t ethx	环回测试。
ethtool -S ethx	显示硬件及驱动统计信息。
ethtool -n ethx	显示当前支持的RSS ( Receive Side Scaling ) 规则。
ethtool -N ethx	设置RSS规则。
ethtool -x ethx	查询间接表。
ethtool -X ethx	设置间接表。
ethtool -P ethx	显示固化MAC。
ethtool -l ethx	显示当前队列个数。
ethtool -L ethx	设置队列个数。
ethtool -m ethx	查询端口光模块信息。
部分ethtool标准命令不适用于IN200，其功能由hinicadm替代。例如：	
<ul style="list-style-type: none"><li>ethtool -e：提供EEPROM烧写操作，一般用于网卡固件升级。在配套IN200时，其功能由hinicadm工具的<a href="#">updatefw</a>命令替代。</li><li>ethtool -d：提供芯片寄存器dump功能，一般用于收集芯片定位信息。在配套IN200时，其功能由hinicadm工具的<a href="#">hilink_dump</a>命令替代。</li></ul>	

常用的ifconfig命令及功能如表5-2所示。关于ifconfig指定命令的详细使用方法，可通过命令行自带的帮助信息获取。

表 5-2 常用 ifconfig 命令

命令	功能
ifconfig ethx hw	设置网卡地址。
ifconfig ethx mtu	设置网口MTU。
ifconfig ethx [-]allmulti	设置全组播特性的启用状态。

命令	功能
ifconfig ethx [-]promisc	设置混杂模式的启用状态。

常用的ip link命令及功能如表5-3所示。关于ip link指定命令的详细使用方法，可通过命令行自带的帮助信息获取。

表 5-3 常用 ip link 命令

命令	功能
ip link set dev ethx vf <num> <mac addr>	设置虚拟网口MAC地址。
ip link set dev ethx vf <num> vlan <vlan_id> qos <vlan-qos>	设置虚拟网口VLAN ID及Qos优先级。 <b>说明</b> 不允许用户将vlan_id设置为4095。
ip link set dev ethx vf <num> <rate txrate>	设置虚拟网口限速参数。
ip link set dev ethx vf <num> state <status>	设置虚拟网口Link状态。

常用的vconfig命令及功能如表5-4所示。关于vconfig指定命令的详细使用方法，可通过命令行自带的帮助信息获取。

表 5-4 常用 vconfig 命令

命令	功能
vconfig add ethx <vlan_id>	添加VLAN子设备。
vconfig rem <vlan-name>	删除VLAN子设备。

常用的dcbtool命令及功能如表5-5所示。关于dcbtool指定命令的详细使用方法，可通过命令行自带的帮助信息获取。

表 5-5 常用 dcbtool 命令

命令	功能
dcbtool sc ethx dcb on/off	打开或关闭DCB ( Data Center Bridging ) 功能。
dcbtool gc ethx dcb	查询DCB状态。
dcbtool gc ethx pg	查询优先级组的设置情况。

命令	功能
dcbtool sc ethx pg pgid:xxx	设置指定优先级映射到的优先级带宽组。
dcbtool sc ethx pg pgpct:x,x,x,x,x,x,x,x	设置各优先级带宽组的带宽比（总和需为100%）。
dcbtool sc ethx pg strict:xxxxxxxx	设置指定优先级为严格优先级。
dcbtool sc ethx pg up2tc:xxxxxxxx	设置各优先级映射到的TC组。
dcbtool gc ethx pfc	查询各优先级的PFC状态。
dcbtool sc ethx pfc pfcup:xxxxxxxx	设置各优先级的PFC启用状态。

# 6 常用配置

- [6.1 配置驱动加载参数](#)
- [6.2 配置SR-IOV](#)
- [6.3 配置QoS](#)
- [6.4 配置RoCE bonding](#)
- [6.5 配置64队列](#)
- [6.6 日志收集](#)
- [6.7 常用BIOS配置](#)

## 6.1 配置驱动加载参数

如果客户在安装驱动后需要进行差异化处理，则需要在安装驱动之前配置加载参数。

### 6.1.1 驱动加载参数说明

参数	权限	默认值	范围	功能	配置说明
g_aeq_len	444	64K	[64K,512K]	AEQ队列深度	不合法则使用默认值
g_ceq_len	444	64K	[64K,1M]	CEQ队列深度	不合法则使用默认值
g_num_ceqe_in_tasklet	444	1024	> 0	ceq软中断中每次处理的CEQE个数	不合法则使用默认值

参数	权限	默认值	范围	功能	配置说明
set_vf_link_state	444	0	[0,2]	设置默认的Vf link策略: 0-默认, 1-强制link up, 2-强制link down	不合法则使用默认值
tx_pending_limit	444	0	[0,255]	Ci聚合个数	参数类型为byte, 在合法范围内
tx_coalescing_time	444	0	[0,255]	CI聚合时间	参数类型为byte, 在合法范围内
g_mtt_page_size	S_IRUGO	0	[0,2]	RDMA参数, 0-4K, 1-8K, 2-16K; 其他值默认4K	RoCE使用
g_mtt_check_type	S_IRUGO	0	[0,1]	RDMA参数, type 0或者1; 其他值默认1	RoCE使用
disable_vf_load	444	0	[0,1]	vf是否在host上加载	参数类型为bool
num_qps	444	0	[0,16]	默认的NIC队列个数	超出芯片能力时使用芯片能力值
poll_weight	444	64	[1,4096]	NAPI单次轮询收包个数	为0则使用默认值, 大于4096的, 则使用4096
qp_pending_limit	444	2	[0,255]	NIC队列中断聚合个数	参数类型为byte
qp_coalesce_timer_cfg	444	32	[0,255]	NIC队列中断聚合时间	参数类型为byte
set_max_wqe_num	444	8	[1,32]	LRO聚合个数	[1,32]则使用加载参数, 否则使用驱动默认

参数	权限	默认值	范围	功能	配置说明
rx_buff	444	2	[2,16]	rx_buffer大小, 以K为单位, 且需要为2的幂次方	不合法则使用默认值
set_lro_timer	444	16	[1,1024]	LRO聚合时间	[1,1024]则使用加载参数, 否则使用驱动默认
set_link_status_follow	444	3	[0,3]	设置link策略	[0,2]使用加载参数, 3表示不设置, 否则不设置link跟随策略
lro_replenish_thld	444	256	[1,2 <sup>32</sup> -1]	rx_poll流程中当wqe消耗个数超过阈值则跳出循环进行buff补充	参数不合适会导致LRO自适应效果差(功能未使用)
l2nic_interrupt_switch	644	1	[0,1]	是否需要开关中断	参数类型为bool
lro_en_status	444	2, 代表不设置	[0,1]	默认使能LRO	[0,1]使用加载参数, 否则使用驱动默认配置

## 6.1.2 配置链路状态跟随的加载参数

默认ifconfig down是不会联动关闭物理链路的, 如果需要配置ifconfig down时关闭物理链路, 则需要配置set\_link\_status\_follow=1。有三种方法, 其中, 方法1和2的为实时使用生效, 即单次生效, 在系统重启后加载参数失效。方法3为永久生效, 在系统重启后仍然保持加载参数。

### 方法 1

**步骤1** 以root用户登录服务器OS, 在OS桌面单击鼠标右键, 选择“Open Terminal”打开命令行。

**步骤2** 执行如下命令配置加载参数值。

```
modprobe hinic set_link_status_follow=1
```

**步骤3** 执行如下命令确认配置结果。

```
cat /sys/bus/pci/drivers/hinic/module/parameters/set_link_status_follow
```

回显“1”说明配置成功。

----结束

## 方法 2

**步骤1** 以**root**用户登录服务器OS，在OS桌面上单击鼠标右键，选择“Open Terminal”打开命令行。

**步骤2** 执行如下命令配置加载参数值。

```
insmod hinic.ko set_link_status_follow=1
```

**步骤3** 执行如下命令确认配置结果。

```
cat /sys/bus/pci/drivers/hinic/module/parameters/set_link_status_follow
```

回显“1”说明配置成功。

----结束

## 方法 3

### 须知

如果要使用加载参数，必须在安装驱动rpm/deb包之前创建配置文件，否则会导致重启后加载参数不生效。

**步骤1** 以**root**用户登录服务器OS，在OS桌面上单击鼠标右键，选择“Open Terminal”打开命令行。

**步骤2** 执行如下命令进入加载参数配置文件所在目录。

```
cd /etc/modprobe.d/
```

**步骤3** 执行如下命令创建配置文件。

```
vi hinic.conf
```

按“i”编辑文件，在文件中添加配置加载参数。

```
options hinic set_link_status_follow=1
```

编辑完成后按“Esc”退出编辑模式，输入“:wq!”保存并退出。

**步骤4** 使加载参数生效。

加载参数生效方式有两种，请按照实际需要进行选择。

- 方法一：  
重启OS。  
**reboot**
- 方法二：
  - a. 执行**lsmod |grep hinic**查看hinic是否已加载。
    - 是，执行**步骤4.b**。

- 否，执行**步骤4.c**。
- b. 执行**rmmmod hinic**卸载当前正在运行的驱动。
- c. 执行**modprobe hinic**加载已安装的驱动。

**步骤5** 执行如下命令确认配置结果。

```
cat /sys/bus/pci/drivers/hinic/module/parameters/set_link_status_follow
```

回显“1”说明配置成功。

----结束

## 6.2 配置 SR-IOV

当IN200在SR-IOV环境中工作时，需要确保内核启用SR-IOV，并确保获得良好的性能。

### 6.2.1 X86 版本下配置 SR-IOV

#### 6.2.1.1 配置 BIOS

下面以2288H V5的BIOS为例，介绍在BIOS中开启SR-IOV功能。

##### 说明

操作之前，请保证服务器启动模式为“UEFI”。

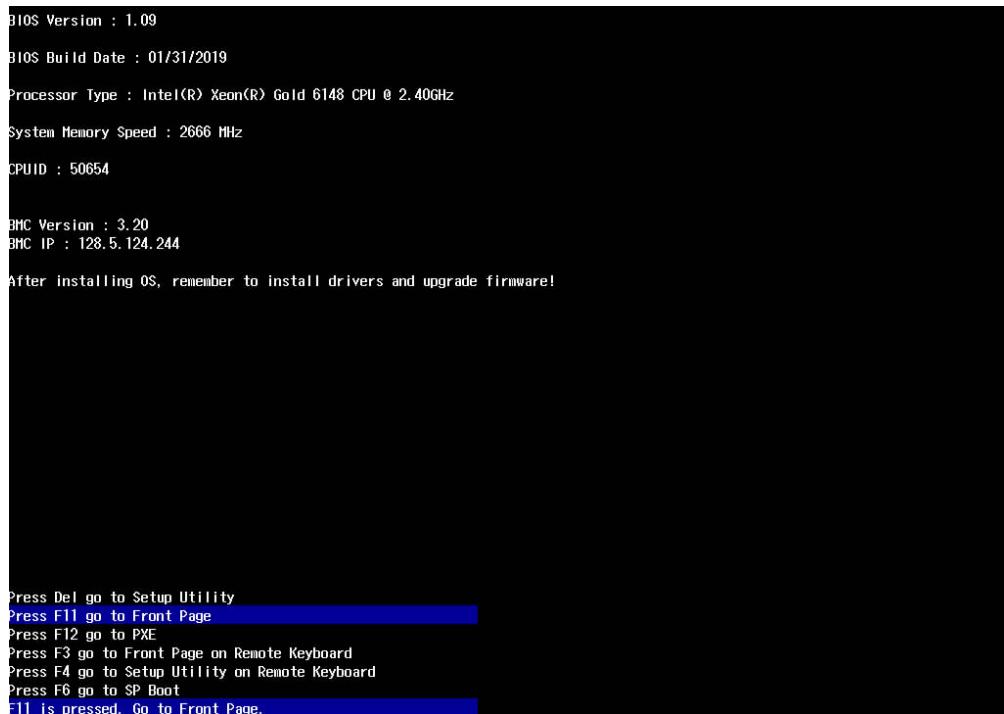
**步骤1** 通过远程虚拟控制台登录服务器实时桌面。

**步骤2** 重启服务器。

**步骤3** 进入BIOS配置界面。

- 当BIOS的版本在V3XX及以下时，界面显示如**图6-1**。

图 6-1 启动信息 (1)



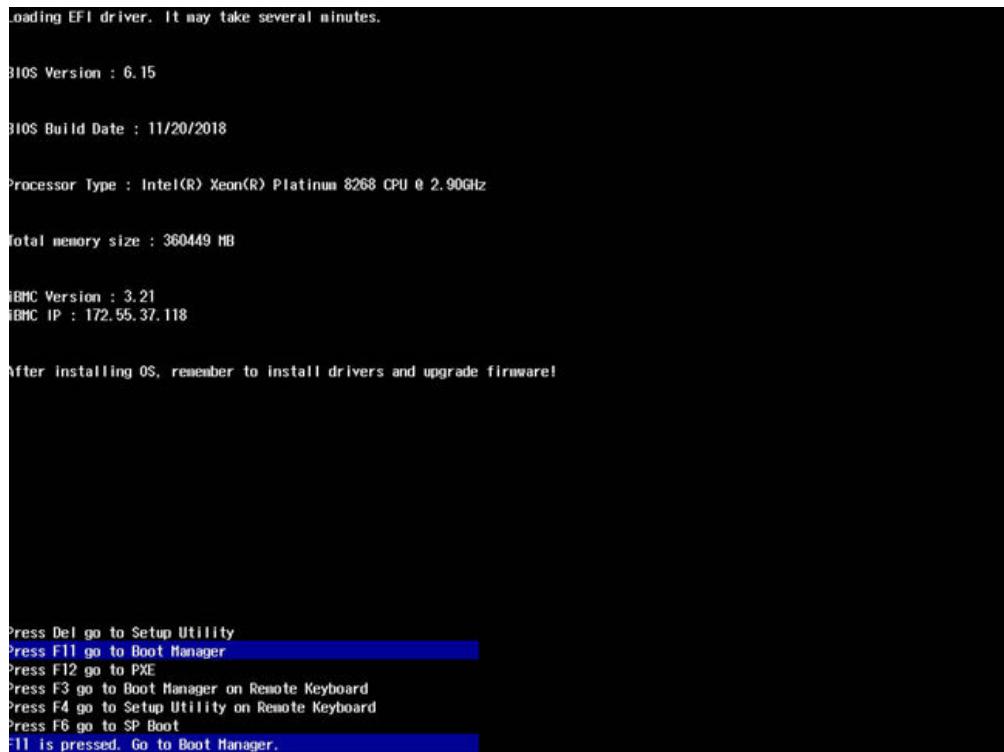
按“F11”进入“BIOS”界面，如图6-2所示，执行步骤5。

图 6-2 “BIOS”界面 (1)



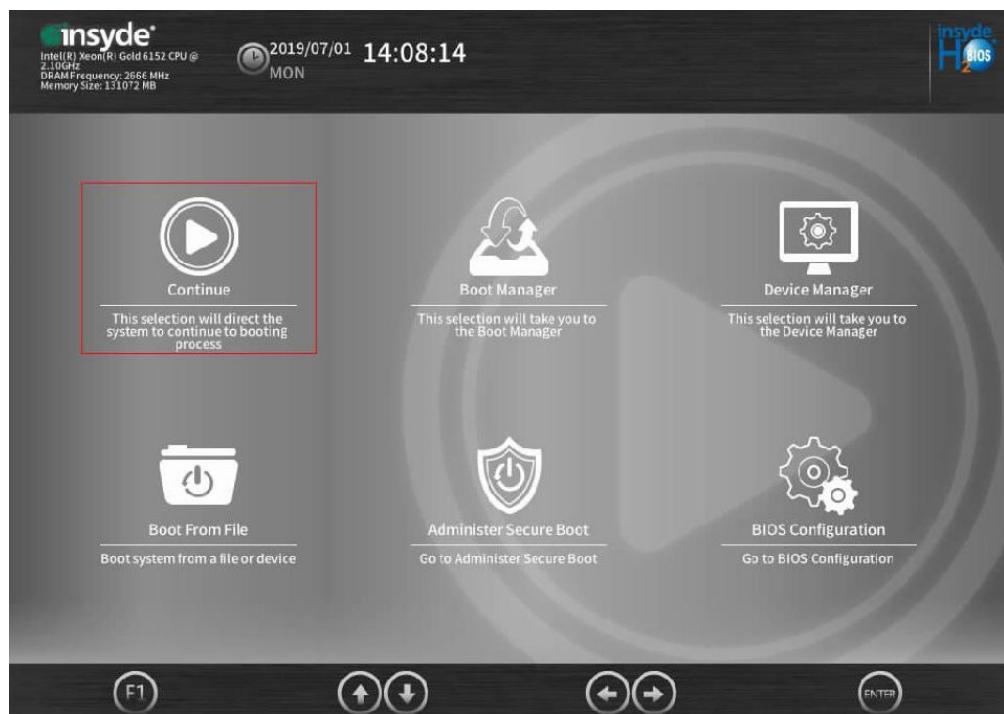
- 当BIOS的版本在V6XX及以上时，界面显示如图6-3。

图 6-3 启动信息 (2)



按“Delete”进入“BIOS”界面，如图6-4所示。

图 6-4 “BIOS”界面 (2)



**步骤4** 用方向键选择右下角的“BIOS Configuration”并按“Enter”，进入“Setup Utility”主界面。执行**步骤6**。

**步骤5** 用方向键选择右下角的“Setup Utility”并按“Enter”，进入“Setup Utility”主界面。

**步骤6** 在左侧导航栏中选择“Advanced”，如图6-5所示。

图 6-5 “Advanced” 主界面

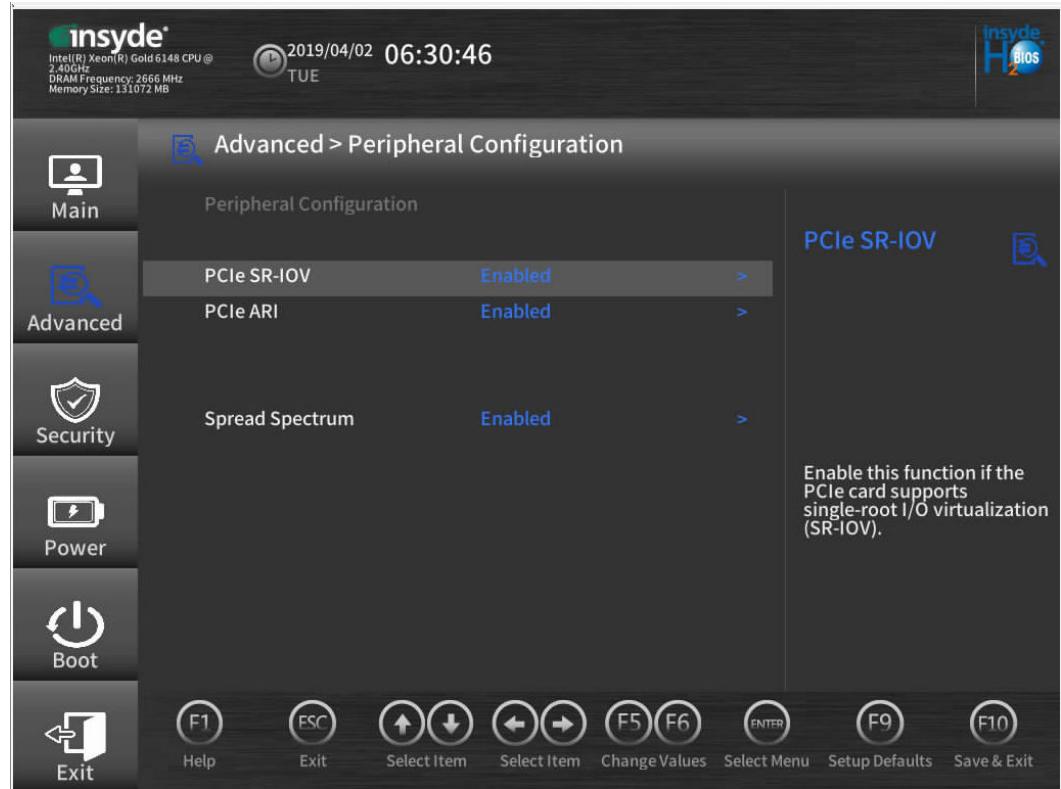


**步骤7** 用方向键选择“Peripheral Configuration”并按“Enter”，进入“Peripheral Configuration”配置界面，如图6-6所示。

设置“PCIe SR-IOV”为“Enabled”状态。

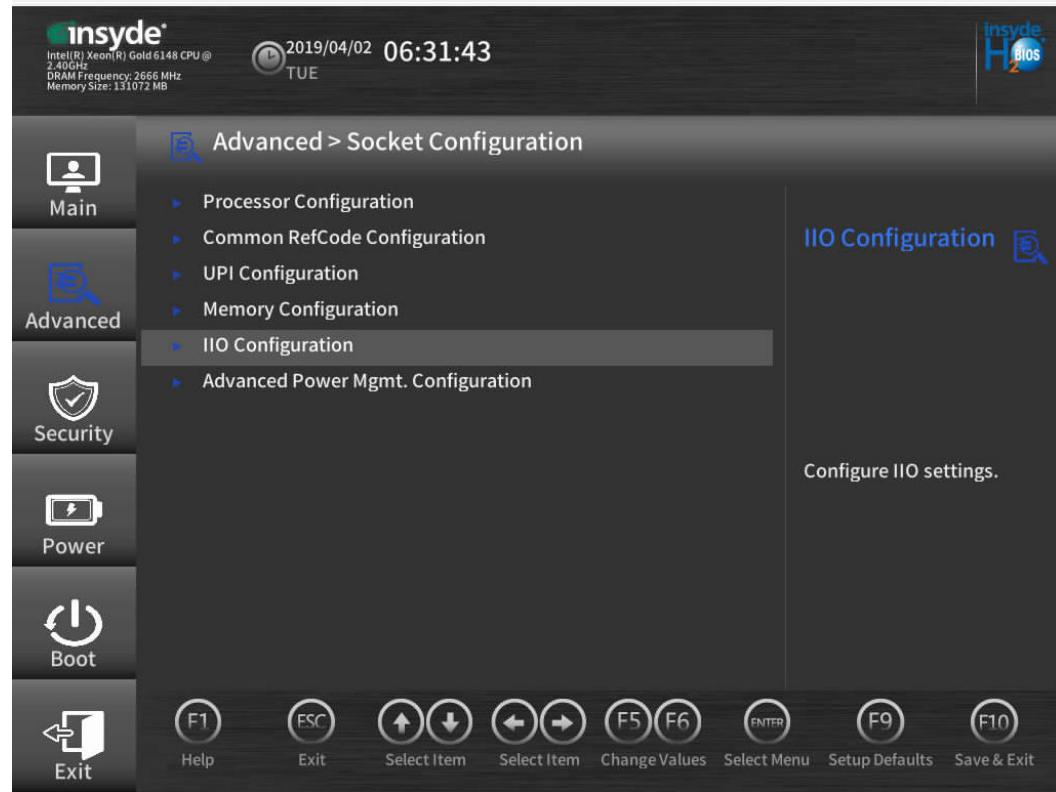
按“Esc”退出并返回“Advanced”主界面。

图 6-6 “Peripheral Configuration” 配置界面



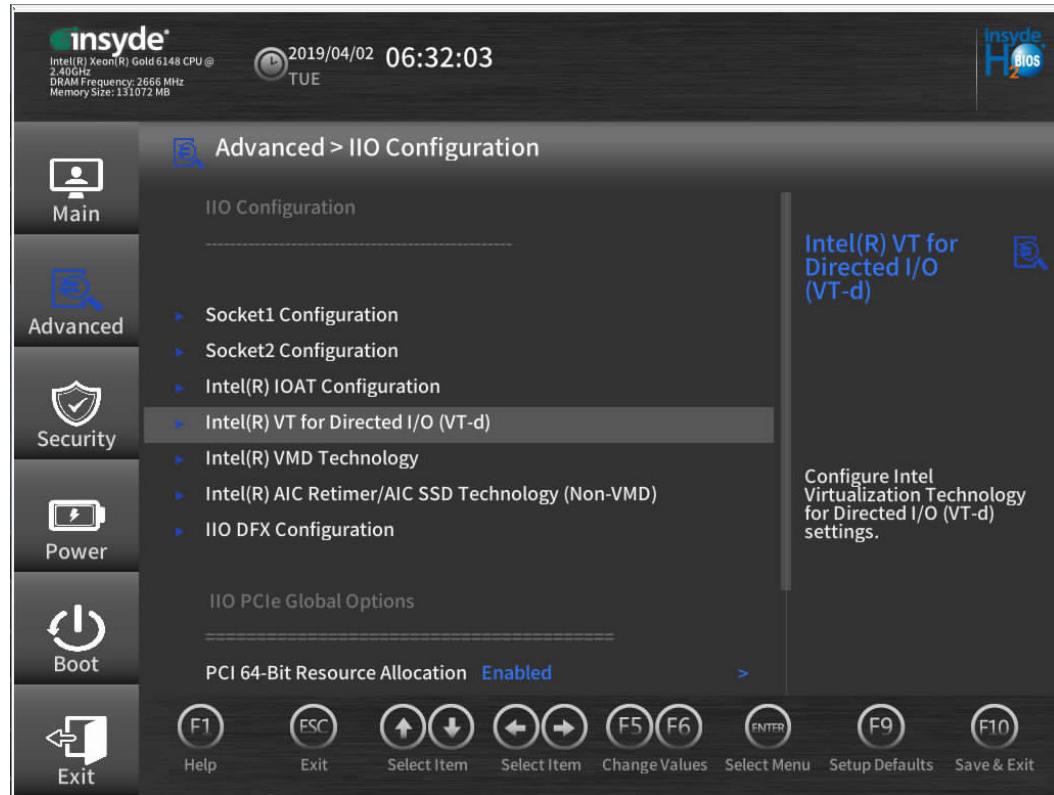
**步骤8** 用方向键选择“Socket Configuration”并按“Enter”，进入“Socket Configuration”配置界面。如图5 “**Socket Configuration**”界面所示。

图 6-7 “Socket Configuration” 配置



**步骤9** 用方向键选择“[IIO Configuration](#)”并按“Enter”，进入“[IIO Configuration](#)”配置界面，如[图6-8所示](#)。

图 6-8 “IIO Configuration” 配置界面



**步骤10** 用方向键选择“Intel(R) VT for Directed I/O (VT-d)”并按“Enter”，进入“Intel(R) VT for Directed I/O (VT-d)”配置界面。如图6-8所示。

设置“Intel(R) VT for Directed I/O (VT-d)”为“Enabled”状态。

按“Esc”退出本界面。

图 6-9 “Intel(R) VT for Directed I/O (VT-d)” 界面



----结束

### 6.2.1.2 修改系统引导配置文件

根据Linux发行版的不同，系统引导配置文件可能是grub.conf或grub.cfg。

本节讨论SR-IOV直通（passthrough，简称pt）模式下配置iommu和intel\_iommu参数。

- 要在内核中启用SR-IOV，请在grub文件中配置intel\_iommu=on。
- 为避免主机侧出现内存映射和性能问题，在使用SR-IOV时，请在grub文件中配置iommu=pt。

### 操作步骤

以RHEL 7.4为例，介绍修改系统引导配置文件的操作步骤。

**步骤1** 执行如下命令确认是否已经使能SR-IOV。

```
cat /proc/cmdline
```

查看回显中是否显示“intel\_iommu=on iommu=pt”。

- 是，操作结束。
- 否，执行**步骤2**。

**步骤2** 执行如下命令在系统中查找对应的grub文件。

```
find /boot/ -name grub.cfg
```

回显如下信息，表示在“/boot/efi/EFI/redhat/”路径下存在grub.cfg文件。

```
/boot/efi/EFI/redhat/grub.cfg
```

**步骤3 编辑grub.cfg文件。**

```
vi /boot/efi/EFI/redhat/grub.cfg
```

按“i”编辑文件，在文件中添加系统启动命令：intel\_iommu=on iommu=pt，添加位置如下图所示。

```
## BEGIN /etc/grub.d/10_Linux ##
menuentry 'Red Hat Enterprise Linux Server (3.10.0-693.el7.x86_64) 7.4 (Maipo)' --class red --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-693.el7.x86_64-advanced-299e0ab2-22c9-42d4-9ff2-dd898bf1d541'
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_gpt
    insmod xfs
    set root='hd0,gpt2'
    if [ $xfEATURE_PLATFORM_SEARCH_HINT = xy ]; then
        search --fs-uuid --set=root --hint-bios=hd0,gpt2 --hint-efi=hd0,gpt2 --hint-baremetal=ahci0,gpt2 91720993-f811-4d2e-8bcc-5dc0fe8e953
    else
        search --no-floppy --fs-uuid --set=root 91720993-f811-4d2e-8bcc-5dc0fe8e953
    fi
    linuxefi /vmlinuz-3.10.0-693.el7.x86_64 root=/dev/mapper/rhel-root ro crashkernel=auto rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet LANG=en_US.UTF-8 intel_iommu=on iommu=pt
    initrdefi /initramfs-3.10.0-693.el7.x86_64.img
menuentry 'Red Hat Enterprise Linux Server (0-rescue-0b0eb4fa67204a6db3a36e22f4084149) 7.4 (Maipo)' --class red --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-0-rescue-0b0eb4fa67204a6db3a36e22f4084149-advanced-299e0ab2-22c9-42d4-9ff2-dd898bf1d541' {
    load_video
}
```

编辑完成后按“Esc”退出编辑模式，输入“:wq!”保存并退出。

**步骤4 执行如下命令重启系统。**

```
reboot
```

**步骤5 执行如下命令查看/proc/cmdline中是否已添加“intel\_iommu=on iommu=pt”成功。**

```
cat /proc/cmdline
```

查看回显中是否显示“intel\_iommu=on iommu=pt”。

- 是，操作结束。
- 否，请重新执行**步骤2~步骤5**。

----结束

## 6.2.2 ARM 版本下配置 SR-IOV

### 6.2.2.1 配置 BIOS

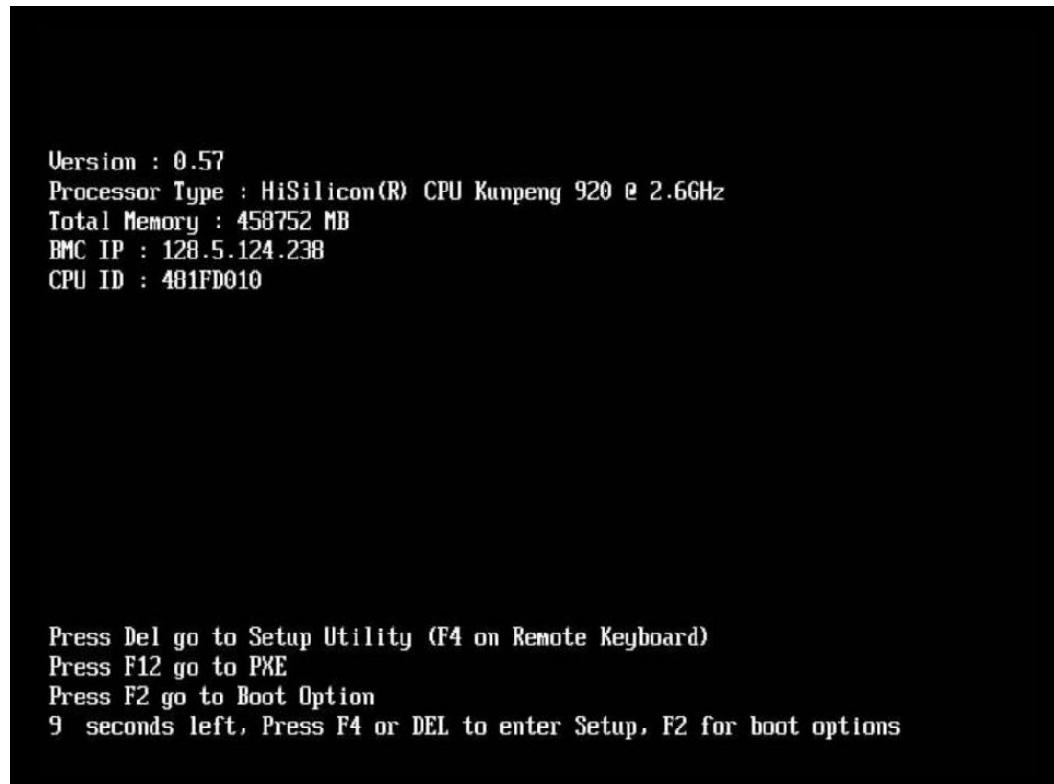
下面以TaiShan 200服务器（型号 2280）的BIOS为例，介绍在BIOS中开启SR-IOV功能。

**步骤1 通过远程虚拟控制台登录服务器实时桌面。**

**步骤2 重启服务器。**

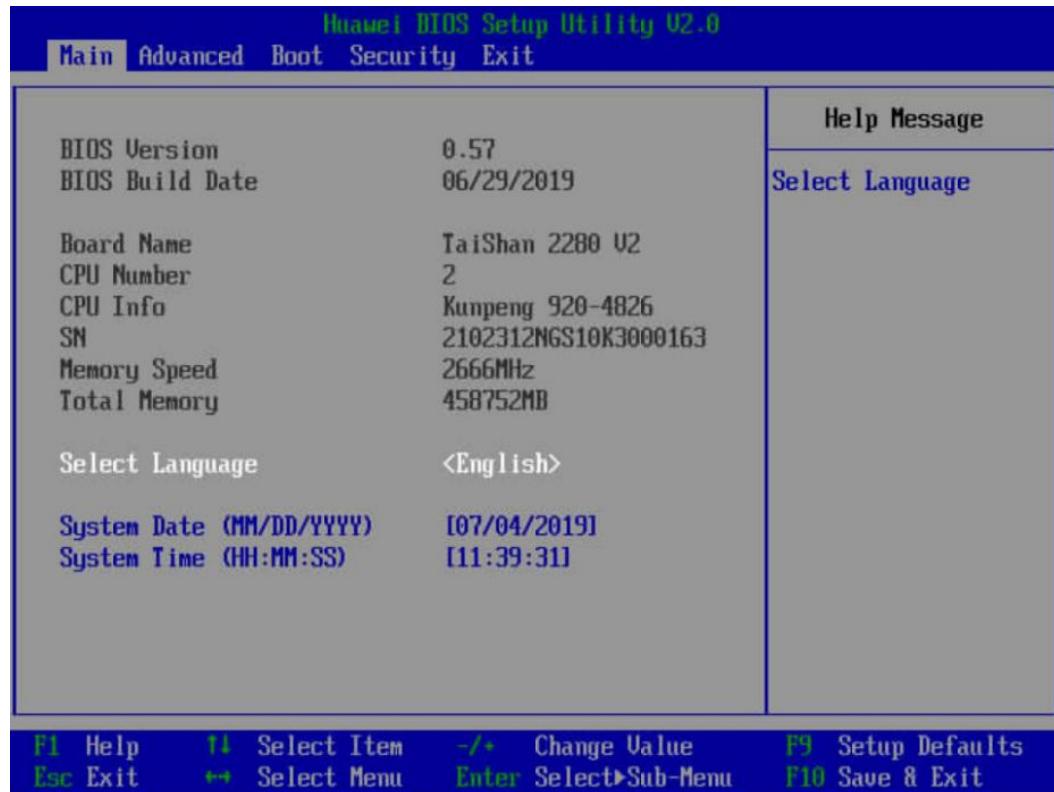
**步骤3 当界面显示如下信息时，按“Delete”，如图6-10所示。**

图 6-10 启动界面



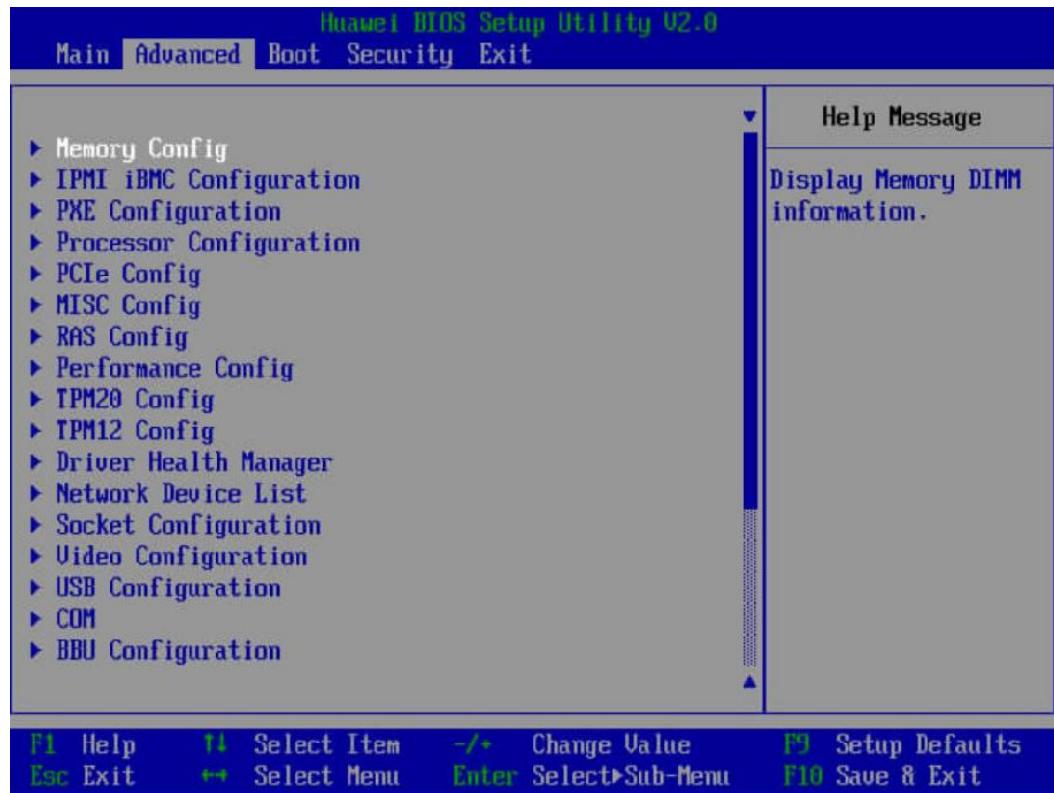
进入“Setup Utility”界面，如图6-11所示。

图 6-11 “Setup Utility”界面



步骤4 用左右方向键选择“Advanced”，进入“Advanced”界面，显示如图6-12。

图 6-12 “Advanced” 主界面



步骤5 用上下方向键选择的“MISC Config”并按“Enter”，进入“MISC Config”配置界面。

步骤6 设置“Support Smmu”为“Enabled”，其余选项保持默认，如图6-13所示。按“F10”退出，BIOS配置完成。

图 6-13 “MISC Config” 界面

MISC Config		Help Message
Support Smmu	<Enabled>	Enable or Disable Smmu.
Support GOP FB for SM750	<Disabled>	
Support SPCR	<Disabled>	
System Debug Level	<Debug>	
Memory Print Level	<Minimum>	
CPU Prefetching	<Enabled>	
Configuration		
Support Down Core	<Disabled>	

F1 Help    F2 Select Item    F3 Change Value    F9 Setup Defaults  
Esc Exit    F4 Select Menu    Enter Select Sub-Menu    F10 Save & Exit

----结束

### 6.2.2.2 修改系统引导配置文件

根据Linux发行版的不同，系统引导配置文件可能是grub.conf或grub.cfg。

为避免主机侧出现内存映射和性能问题，在使用SR-IOV时，请在grub文件中配置iommu.passthrough=1。

本节讨论SR-IOV直通（passthrough，简称pt）模式下配置iommu.passthrough=1。

#### 操作步骤

以RHEL 7.6为例，介绍修改系统引导配置文件的操作步骤。

**步骤1** 执行以下命令，确认是否已经使能SR-IOV。

**cat /proc/cmdline**

查看回显中是否显示“iommu.passthrough=1”。

- 是，操作结束。
- 否，执行**步骤2**。

**步骤2** 执行如下命令在系统中查找对应的grub文件。

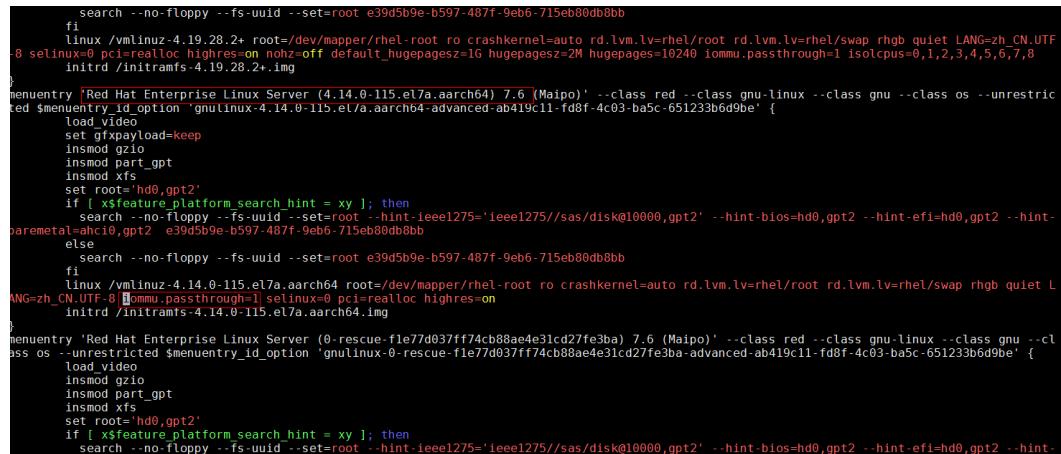
**find /boot/ -name grub.cfg**

回显如下信息，表示在“/boot/efi/EFI/redhat/”路径下存在grub.cfg文件。

/boot/efi/EFI/redhat/grub.cfg

**步骤3 编辑grub.cfg文件。****vi /boot/efi/EFI/redhat/grub.cfg**

按“i”编辑文件，在文件末端添加系统启动命令：iommu.passthrough=1，添加位置如下图所示。



```
search --no-floppy --fs-uuid --set=root e39d5b9e-b597-487f-9eb6-715eb80db8bb
    linux /vmlinuz-4.19.28.2+ root=/dev/mapper/rhel-root ro crashkernel=auto rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet LANG=zh_CN.UTF-8 selinux=0 pci=realloc highres=on nohz=off default_hugepagesz=1G hugepages=10240 iommu.passthrough=1 isolcpus=0,1,2,3,4,5,6,7,8
    initrd /initramfs-4.19.28.2+.img
}
menuentry 'Red Hat Enterprise Linux Server (4.14.0-115.el7a.aarch64) 7.6 (Maipo)' --class red --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-4.14.0-115.el7a.aarch64-advanced-ab419c11-fd8f-4c03-ba5c-651233b6d9be' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_gpt
    insmod xfs
    set root='hd0,gpt2'
    if [ $feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root -hintValue=1275='ieee1275//sas/disk@10000,gpt2' --hint-bios=hd0,gpt2 --hint-efi=hd0,gpt2 --hint-pci=ahci0,gpt2 e39d5b9e-b597-487f-9eb6-715eb80db8bb
    else
        search --no-floppy --fs-uuid --set=root e39d5b9e-b597-487f-9eb6-715eb80db8bb
    fi
    linux /vmlinuz-4.14.0-115.el7a.aarch64 root=/dev/mapper/rhel-root ro crashkernel=auto rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet LANG=zh_CN.UTF-8 iommu.passthrough=1 selinux=0 pci=realloc highres=on
    initrd /initramfs-4.14.0-115.el7a.aarch64.img
}
menuentry 'Red Hat Enterprise Linux Server (0-rescue-fle7d037ff74cb88ae4e31cd27fe3ba) 7.6 (Maipo)' --class red --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-0-rescue-fle7d037ff74cb88ae4e31cd27fe3ba-advanced-ab419c11-fd8f-4c03-ba5c-651233b6d9be' {
    load_video
    insmod gzio
    insmod part_gpt
    insmod xfs
    set root='hd0,gpt2'
    if [ $feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root -hintValue=1275='ieee1275//sas/disk@10000,gpt2' --hint-bios=hd0,gpt2 --hint-efi=hd0,gpt2 --hint-pci=ahci0,gpt2 e39d5b9e-b597-487f-9eb6-715eb80db8bb
    else
        search --no-floppy --fs-uuid --set=root e39d5b9e-b597-487f-9eb6-715eb80db8bb
    fi
}
```

编辑完成后按“Esc”退出编辑模式，输入“:wq!”保存并退出。

**步骤4 执行如下命令重启系统。****reboot****步骤5 执行如下命令查看/proc/cmdline中是否已添加“iommu.passthrough=1”成功。****cat /proc/cmdline**

查看回显中是否显示“iommu.passthrough=1”。

- 是，操作结束。
- 否，请重新执行**步骤2~步骤5**。

**----结束**

## 6.3 配置 QoS

### 6.3.1 支持流控概述

IN200支持PAUSE帧的流控和PFC的流控，初始状态使能PAUSE流控，通过**ethtool -a DEV\_NAME**命令可以查询PAUSE帧流控的使能状态；当PFC使能时，PAUSE帧流控会被关闭。

根据网络流量分类方式的不同，PFC流控可分为基于VLAN tag的流控和基于DSCP的流控，当PFC使能时，默认情况使用基于VLAN tag的流控。

### 6.3.2 基于 VLAN tag 的流控配置方法

#### 前提条件

- 已完成IN200网卡的RoCE驱动包的安装。

- 已完成交换机和网卡的VLAN配置。

## 配置方法

配置基于VLAN tag的QoS需要依次执行以下操作：

- 设置QoS。设置QoS通过hinicadm管理工具来实现的，具体命令介绍请参见[4.4.37 查询和设置用户优先级、cqe水线 \( qos \)](#)。
- 设置DCB。设置DCB通过hinicadm管理工具来实现的，具体命令介绍请参见[4.4.10 查询和设置DCB功能 \( dcb \)](#)。
- 设置PFC。设置PFC通过hinicadm管理工具来实现的，具体命令介绍请参见[4.4.36.1 查询PFC功能](#)。

### 6.3.3 基于 DSCP 的流控配置方法

#### 前提条件

- 已完成IN200网卡的RoCE驱动包的安装。
- 已完成交换机的DSCP配置。

#### 约束条件

IN200网卡在使能ROCE\_DSCP功能后，所有端口都会使能DSCP功能，并且不支持和VLAN tag优先级共存（若存在VLAN设备，端到端映射还是使用DSCP映射）。

## 配置方法

配置基于DSCP的QoS需要依次执行以下操作：

- 设置QoS。设置QoS通过hinicadm管理工具来实现的，具体命令介绍请参见[4.4.37 查询和设置用户优先级、cqe水线 \( qos \)](#)。
- 设置DCB。设置DCB通过hinicadm管理工具来实现的，具体命令介绍请参见[4.4.10 查询和设置DCB功能 \( dcb \)](#)。
- 设置PFC。设置PFC通过hinicadm管理工具来实现的，具体命令介绍请参见[4.4.36.1 查询PFC功能](#)。
- 设置ROCE\_DSCP。设置ROCE\_DSCP通过hinicadm管理工具来实现的，具体命令介绍请参见[4.4.41 配置ROCE\\_DSCP功能 \( roce\\_dscp \)](#)。

## 6.4 配置 RoCE bonding

RoCE bonding是以OS提供的bonding功能为基础，结合RoCE驱动，通过切换物理端口来实现bonding的一种机制。通过将多个物理端口绑定为一个逻辑端口的方式实现容错、带宽扩容和负载均衡的功能。

### 6.4.1 RoCE bonding 概述

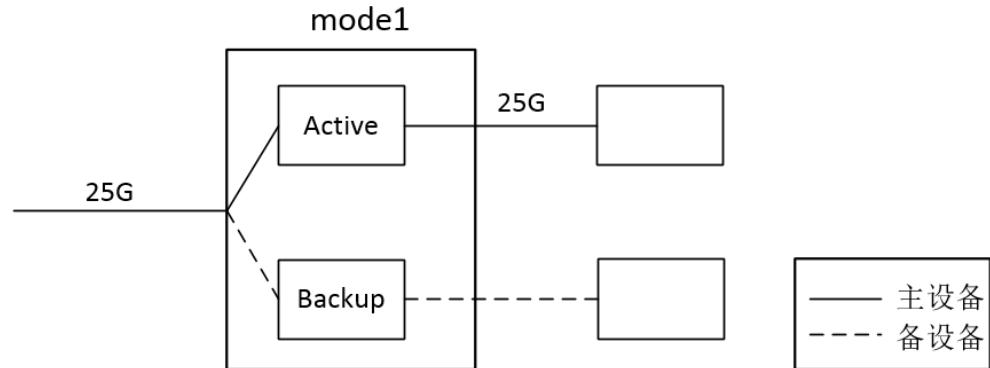
RoCE bonding支持Active-backup模式1、Balance-XOR模式2、802.3ad模式4三种模式。

- Active-backup mode1：主备模式，只有一个设备处于活动状态，当一个设备宕掉后另一个设备由备设备转换为主设备。MAC地址是外部可见的。从外部来看，

bonding的MAC地址是唯一的，以避免交换机发生混乱。此模式只提供了容错能力，可以提供高网络连接的可用性，但资源利用率比较低。

Active-backup模式1如图6-14所示。

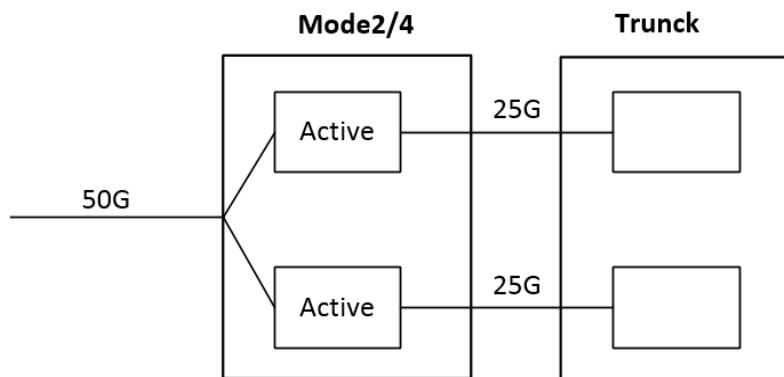
图 6-14 Active-backup mode1



- Balance-XOR mode2: 平衡模式，基于所选择的Hash策略传输数据，本模式提供负载均衡和容错能力。
- 802.3ad mode4: IEEE 802.3ad动态链路聚合模式，创建一个聚合组，它们共享同样的速率和双工设定。根据802.3ad规范将多个设备工作在同一个活动的聚合体下。

Balance-XOR模式2和802.3ad模式4如图6-15所示。

图 6-15 Balance-XOR mode2/802.3ad mode4



## 6.4.2 RoCE bonding 支持的操作系统

表 6-1 RoCE 支持的操作系统

操作系统	x86	ARM
BC-Linux	-	7.6

操作系统	x86	ARM
CentOS	7.4/7.5/7.6/8.0/8.0_ini/8.1/8.2/8.2_ini <b>说明</b> CentOS 8.X_ini使用OS自带OFED，无需执行安装OFED的步骤。	7.5/7.6/8.0/8.1/8.2
EulerOS	V200R005C00SPC310B05 6/ V200R007C00SPC500B00 5	V200R008C00SPC106B350/ V200R008C00SPC306B830/ V200R009C00SPC100B150
NeoKylin ( 中标麒麟 )	-	7.6
OpenEuler	-	20.3
RHEL	7.5/7.6/8.0	-
SUSE	12.2/12.4	12.4/15.1
Ubuntu	16.04.4/18.04.1	18.04.1/18.04.2/18.04.3
UOS	-	20SP1

### 6.4.3 配置方法

Bonding可以通过命令行和配置文件的两种方式实现bonding设备的创建、销毁和配置等操作，不推荐使用NetworkManager配置bonding。

#### 前提条件

已完成IN200网卡的RoCE驱动包的安装。

已完成VLAN配置。

#### 注意事项

- RoCE bonding只支持IN200同一个网卡网口的绑定。
- 有多种类型的IN200网卡的环境中，不支持2个以上网口绑定（如2\*25与4\*25G的IN200网卡混合环境中，不支持同一张卡2个以上网口的绑定。）

#### 6.4.3.1 通过命令行配置 RoCE bonding

以CentOS 7.4为例，使用命令行进行bonding配置。

##### 须知

该方式为临时生效，OS下复位网络服务或OS重启后配置会丢失。

**步骤1** 以root用户登录服务器OS，在OS桌面单击鼠标右键，选择“Open Terminal”打开命令行。

**步骤2** 执行如下命令找到up的端口。

**ibdev2netdev**

**步骤3**（可选）如果进行2个以上网口bonding时，需要修改/etc/modprobe.d/hiroce\_pro.conf配置文件，以4\*25的卡进行4个网口绑定为例：

1. 执行以下命令打开/etc/modprobe.d/hiroce\_pro.conf配置文件：

**vi /etc/modprobe.d/hiroce\_pro.conf**

按“i”编辑文件，文件内容如下：

```
options hiroce g_want_bond_slave_cnt=0x4 g_want_bond0_slave_bits=0xf  
g_want_bond1_slave_bits=0x0
```

- g\_want\_bond\_slave\_cnt表示要绑定的网口数量，4\*25G的4个网口此值为0x4。
- g\_want\_bond0\_slave\_bits表示bond0使用的网口的比特位，4\*25G的4个网口此值为0xf。
- g\_want\_bond1\_slave\_bits表示bond1使用的网口的比特位，4\*25G的4个网口绑定只能绑定成一个bond0，因此此值为0x0。

编辑完成后按“Esc”退出编辑模式，输入“:wq!”保存并退出。

2. 执行以下命令使hiroce\_pro.conf文件生效。

**service openibd restart**

**步骤4** 执行如下命令，以enp6s0和enp7s0端口为例，创建“bond0”。

**ip link add bond0 type bond mode X**

#### □ 说明

- X取值为1、2、4。
- 命令式中通过X的值设置相应的bonding模式。
- 若mode设置为1，则交换机正常配置；若mode设置为2，则交换机需要配Trunk；若mode设置为4，则交换机需配置Trunk为静态LACP模式，速率设置推荐与bonding对应。

**步骤5**（可选）若mode设置为4，则推荐配置lacp\_rate，执行如下命令配置lacp\_rate。否则，执行**步骤6**。

**echo 1 > /sys/devices/virtual/net/bond0/bonding/lacp\_rate**

**步骤6** 执行如下命令使能bond0。

**ifconfig bond0 up**

**步骤7** 执行如下命令对“bond0”使用的enp6s0和enp7s0网口进行bonding。

**ifenslave bond0 enp6s0 enp7s0**

**步骤8** 执行如下命令设置网络的检查周期。

**echo 100 > /sys/class/net/bond0/bonding/miimon #miimon表示多长时间检查一次网络，单位ms。**

**步骤9** 执行如下命令为“bond0”创建IP。

```
ifconfig bond0 /P #IP为bond0配置的IP地址
```

----结束

### 6.4.3.2 通过配置文件配置 RoCE bonding

以CentOS 7.4为例，使用配置文件进行bonding配置步骤如下：

**步骤1** 以root用户登录服务器OS，在OS桌面单击鼠标右键，选择“Open Terminal”打开命令行。

**步骤2** 执行如下命令找到up的端口。

```
ibdev2netdev
```

**步骤3** 执行如下命令打开/etc/sysconfig/network-scripts目录。

```
cd /etc/sysconfig/network-scripts
```

**步骤4** 执行如下命令创建并打开配置文件“ifcfg-fsb\_bond”。

```
vi ifcfg-fsb_bond
```

按“i”编辑文件，文件内容如下：

```
DEVICE=fsb_bond
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
TYPE=Bonding
MTU=1500
BONDING_OPTS='mode=1 miimon=100 updelay=0 downdelay=0 num_grat_arp=50'
NM_CONTROLLED=no
IPADDR=
NETMASK=
NETWORK=
```

#### 说明

- BONDING\_OPTS配置项中设置bonding的模式和检查网络的时间。
- 若mode设置为1，则交换机正常配置；若mode设置为2，则交换机需要配Trunk；若mode设置为4，则交换机需配置Trunk为静态LACP模式，速率设置推荐与bonding对应。
- 若在bonding的配置文件BONDING\_OPTS项中设置bonding的模式为4，则推荐设置lacp\_rate。

编辑完成后按“Esc”退出编辑模式，输入“:wq!”保存并退出。

**步骤5** 以enp8s0和enp9s0端口为例进行fsb\_bond绑定，在/etc/sysconfig/network-scripts目录下分别修改“ifcfg-enp8s0”和“ifcfg-enp9s0”，添加选项“MASTER=fsb\_bond”。

以编辑“ifcfg-enp8s0”文件为例，执行如下命令：

```
vi ifcfg-enp8s0
```

按“i”编辑文件，文件内容如下：

```
DEVICE=enp8s0
BOOTPROTO=none
ONBOOT=yes
MASTER=fsb_bond
SLAVE=yes
USERCTL=no
NM_CONTROLLED=no
```

编辑完成后按“Esc”退出编辑模式，输入“:wq!”保存并退出。

**步骤6**（可选）如果进行2个以上网口bonding时，需要修改/etc/modprobe.d/hiroce\_pro.conf配置文件，以4\*25的卡进行4个网口绑定为例：

1. 执行以下命令打开/etc/modprobe.d/hiroce\_pro.conf配置文件：

```
vi /etc/modprobe.d/hiroce_pro.conf
```

按“i”编辑文件，文件内容如下：

```
options hiroce g_want_bond_slave_cnt=0x4 g_want_bond0_slave_bits=0xf  
g_want_bond1_slave_bits=0x0
```

- g\_want\_bond\_slave\_cnt表示要绑定的网口数量，4\*25G的4个网口此值为0x4。
- g\_want\_bond0\_slave\_bits表示bond0使用的网口的比特位，4\*25G的4个网口此值为0xf。
- g\_want\_bond1\_slave\_bits表示bond1使用的网口的比特位，4\*25G的4个网口绑定只能绑定成一个bond0，因此此值为0x0。

编辑完成后按“Esc”退出编辑模式，输入“:wq!”保存并退出。

2. 执行以下命令使hiroce\_pro.conf文件生效。

```
service openibd restart
```

**步骤7** 执行以下命令使配置生效。

### 须知

执行此命令会重启操作系统下的网络服务，所有网卡网络会短暂失去连接，有可能导致业务中断，建议在业务空闲时段进行操作。

```
service network restart
```

----结束

## 6.5 配置 64 队列

当前仅对SP570/SP580/SP572支持64队列配置，其他卡不支持。

### 6.5.1 配置 4\*25GE 网卡 64 队列

本章节的配置方法仅对SP570/SP580有效，其他卡不支持。

#### 6.5.1.1 概述

SP570/SP580默认情况下每个PF支持16队列，每个PF支持的VF是60个，VF队列数是4个。若需配置每个PF支持64队列，可通过hinicconfig工具配合配置模板进行配置，每个PF的VF数修改为24，VF的队列数修改为8。

#### 说明

当VF队列数为4个时，单VF无法达到100GE卡极限带宽。

### 6.5.1.2 配置方法

#### 前提条件

- IN200的整体驱动包已上传到服务器OS。  
NIC驱动包及hinicconfig工具包含在软件包中，获取方法请参见[3.1 获取软件包](#)。
- 服务器OS上已安装了IN200对应的NIC驱动包。  
安装方法请参见[3.2.1 NIC驱动维护](#)。

#### 操作步骤

**步骤1** 登录服务器OS。

**步骤2** 在操作系统命令行执行如下命令进入驱动包解压后的目录，例如：/home。

```
cd /home
```

**步骤3** 执行如下命令查询当前的配置模板。

```
cd tools/linux/nic/config
```

```
./hinicconfig hinico
```

- 若回显如下信息，说明设备没有配置过模板，执行**步骤4**。

```
Cannot find template in card, maybe it's empty.  
Note: the template is empty by default.
```

- 若回显如下信息，说明设备配置过模板，执行**步骤5**。

```
Device current configuration:
```

```
[BASIC]
    MODENAME: S425_GW_NIC_4PF
    HN: 0
    PN: 3
    PM: 0x000F
    PH: 0x00000000
    INP: 7
    VINP: 4

[VNP]
PF12 PF13 PF14 PF15
  24   24   24   24
  00   00   00   00
  00   00
[QNP]
PF12 PF13 PF14 PF15
  63   63   63   63
  00   00   00   00
  00   00
[VQNP]
PF12 PF13 PF14 PF15
  07   07   07   07
  00   00   00   00
  00   00
[QPNP]
PF12 PF13 PF14 PF15
  131072 131072 131072 131072
  00   00   00   00
  00   00
[VQPNP]
PF12 PF13 PF14 PF15
  1024 1024 1024 1024
  00   00   00   00
  00   00
[CQNP]
PF00 PF01 PF02 PF03 PF04 PF05 PF06 PF07 PF08 PF09 PF10 PF11
  00   00   00   00   00   00   00   00   00   00   00   00
```

```
PF12  PF13  PF14  PF15
      131072 131072 131072 131072 00    00    00    00    00    00    00    00
00    00    00
[VCQNP]
PF00  PF01  PF02  PF03  PF04  PF05  PF06  PF07  PF08  PF09  PF10  PF11
PF12  PF13  PF14  PF15
      131072 8192  8192  16384 00    00    00    00    00    00    00    00
00    00    00
```

- 若回显其他信息，则说明设备配置过模板但配置异常，执行[步骤4](#)。

#### 步骤4 执行如下命令配置64队列。

```
./hinicconfig hinic0 -f std_sh_4x25ge_dpdk_cfg_template0.ini
```

回显如下信息：

```
linux-gIRJWR:/home/tools/linux/nic/config # ./hinicconfig hinic0 -f std_sh_4x25ge_dpdk_cfg_template0.ini
Template std_sh_4x25ge_dpdk_cfg_template0.ini will be configured to device hinic0
Enter [yes] to continue:
```

按提示输入“**yes**”，回显如下信息：

```
#####
#####
# WARNING! This operation will modify the default configuration. The config  #
# file must be confirmed by Hi1822 manufacturer. DO NOT modify it without  #
# permission. Otherwise, it may cause unrecoverable consequences.          #
#####
#
Enter [agree] to continue:
```

按提示输入“**agree**”，回显如下信息：

```
Succeed to configure device hinic0
```

表示配置成功。

#### 步骤5 重启服务器生效。

**reboot**

----结束

### 6.5.1.3 恢复默认配置

如果不再需要64队列，则需要将配置恢复为默认情况。

#### 前提条件

- IN200的整体驱动包中已上传到服务器OS。  
NIC驱动包及hinicconfig工具包含在软件包中，获取方法请参见[3.1 获取软件包](#)。
- 服务器OS上已安装了IN200对应的NIC驱动包。  
安装方法请参见[3.2.1 NIC驱动维护](#)。

#### 操作步骤

##### 步骤1 登录服务器OS。

##### 步骤2 在操作系统命令行执行如下命令进入驱动包解压后的目录，例如：/home。

```
cd /home
```

**步骤3** 执行如下命令查询当前的配置模板。

```
cd tools/linux/nic/config
```

```
./hinicconfig hinic0
```

- 若设备没有配置过模板，则回显如下信息：

```
Cannot find template in card, maybe it's empty.  
Note: the template is empty by default.
```

- 若设备配置过模板，则回显如下信息：

```
Device current configuration:
```

```
[BASIC]
    MODENAME: S425_GW_NIC_4PF
    HN: 0
    PN: 3
    PM: 0x000F
    PH: 0x00000000
    INP: 7
    VINP: 4

[VNP]
    PF00  PF01  PF02  PF03  PF04  PF05  PF06  PF07  PF08  PF09  PF10  PF11
PF12  PF13  PF14  PF15
    24    24    24    24    00    00    00    00    00    00    00    00
00    00

[QNP]
    PF00  PF01  PF02  PF03  PF04  PF05  PF06  PF07  PF08  PF09  PF10  PF11
PF12  PF13  PF14  PF15
    63    63    63    63    00    00    00    00    00    00    00    00
00    00

[VQNP]
    PF00  PF01  PF02  PF03  PF04  PF05  PF06  PF07  PF08  PF09  PF10  PF11
PF12  PF13  PF14  PF15
    07    07    07    07    00    00    00    00    00    00    00    00
00    00

[QPNP]
    PF00  PF01  PF02  PF03  PF04  PF05  PF06  PF07  PF08  PF09  PF10  PF11
PF12  PF13  PF14  PF15
    131072 131072 131072 131072 00    00    00    00    00    00    00    00
00    00    00

[VQPNP]
    PF00  PF01  PF02  PF03  PF04  PF05  PF06  PF07  PF08  PF09  PF10  PF11
PF12  PF13  PF14  PF15
    1024  1024  1024  1024  00    00    00    00    00    00    00    00
00    00    00

[CQNP]
    PF00  PF01  PF02  PF03  PF04  PF05  PF06  PF07  PF08  PF09  PF10  PF11
PF12  PF13  PF14  PF15
    131072 131072 131072 131072 00    00    00    00    00    00    00    00
00    00    00

[VCQNP]
    PF00  PF01  PF02  PF03  PF04  PF05  PF06  PF07  PF08  PF09  PF10  PF11
PF12  PF13  PF14  PF15
    131072 8192   8192   16384 00    00    00    00    00    00    00    00
00    00    00
```

**步骤4** 执行如下命令恢复默认配置。

```
./hinicconfig hinic0 -f std_sh_4x25ge_nic_cfg_template0.ini
```

回显如下信息：

```
linux-gIRJWR:/home/tools/linux/nic/config # ./hinicconfig hinic0 -f std_sh_4x25ge_nic_cfg_template0.ini
Template std_sh_4x25ge_nic_cfg_template0.ini will be configured to device hinic0
Enter [yes] to continue:
```

按提示输入“**yes**”，回显如下信息：

```
#####
#####
```

```
# WARNING! This operation will modify the default configuration. The config    #
# file must be confirmed by Hi1822 manufacturer. DO NOT modify it without    #
# permission. Otherwise, it may cause unrecoverable consequences.          #
#####
##### Enter [agree] to continue:
```

按提示输入“**agree**”，回显如下信息：

```
Succeed to configure device hinic0
```

表示配置成功。

**步骤5** 重启服务器生效。

**reboot**

----结束

## 6.5.2 配置 2\*100GE 网卡 64 队列

本章节的配置方法仅对SP572有效，其他卡不支持。

### 6.5.2.1 概述

SP572默认情况下每个PF支持16队列，每个PF支持的VF是60个，VF队列数是4个。若需配置每个PF支持64队列，可通过hinicconfig工具配合配置模板进行配置。

### 6.5.2.2 配置方法

#### 前提条件

- IN200的整体驱动包已上传到服务器OS。  
NIC驱动包及hinicconfig工具包含在软件包中，获取方法请参见[3.1 获取软件包](#)。
- 服务器OS上已安装了IN200对应的NIC驱动包。  
安装方法请参见[3.2.1 NIC驱动维护](#)。

#### 操作步骤

**步骤1** 登录服务器OS。

**步骤2** 在操作系统命令行执行如下命令进入驱动包解压后的目录，例如：/home。

```
cd /home
```

**步骤3** 执行如下命令查询当前的配置模板。

```
cd tools/linux/nic/config
```

```
./hinicconfig hinic0
```

- 若回显如下信息，说明设备没有配置过模板，执行**步骤4**。  
Cannot find template in card, maybe it's empty.  
Note: the template is empty by default.

- 若回显如下信息，说明设备配置过模板，执行**步骤5**。

```
Device current configuration:  
[BASIC]  
MODENAME: S2100_SRV_NIC_3PF  
HN: 0
```

```
PN: 2
PM: 0x0007
PH: 0x00000000
INP: 7
VINP: 4
[VNP]
PF00 PF01 PF02 PF03 PF04 PF05 PF06 PF07 PF08 PF09 PF10 PF11 PF12 PF13 PF14 PF15
60 60 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[QNP]
PF00 PF01 PF02 PF03 PF04 PF05 PF06 PF07 PF08 PF09 PF10 PF11 PF12 PF13 PF14 PF15
63 63 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[VQNP]
PF00 PF01 PF02 PF03 PF04 PF05 PF06 PF07 PF08 PF09 PF10 PF11 PF12 PF13 PF14 PF15
03 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[QPNP]
PF00 PF01 PF02 PF03 PF04 PF05 PF06 PF07 PF08 PF09 PF10 PF11 PF12 PF13 PF14 PF15
131072 131072 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[VQPNP]
PF00 PF01 PF02 PF03 PF04 PF05 PF06 PF07 PF08 PF09 PF10 PF11 PF12 PF13 PF14 PF15
1024 1024 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[CQNP]
PF00 PF01 PF02 PF03 PF04 PF05 PF06 PF07 PF08 PF09 PF10 PF11 PF12 PF13 PF14 PF15
131072 131072 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[VCQNP]
PF00 PF01 PF02 PF03 PF04 PF05 PF06 PF07 PF08 PF09 PF10 PF11 PF12 PF13 PF14 PF15
131072 131072 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

- 若回显其他信息，则说明设备配置过模板但配置异常，执行**步骤4**。

#### 步骤4 执行如下命令配置64队列。

```
./hinicconfig hinic0 -f std_sh_2x100ge_nic_cfg_64_queues_template3pf.ini
```

回显如下信息：

```
linux-gIRJWR:/home/tools/linux/nic/config # ./hinicconfig hinic0 -f
std_sh_2x100ge_nic_cfg_64_queues_template3pf.ini
Template std_sh_2x100ge_nic_cfg_64_queues_template3pf.ini will be configured to device hinic0
Enter [yes] to continue:
```

按提示输入“**yes**”，回显如下信息：

```
#####
#####
# WARNING! This operation will modify the default configuration. The config   #
# file must be confirmed by Hi1822 manufacturer. DO NOT modify it without   #
# permission. Otherwise, it may cause unrecoverable consequences.      #
#####
#####
Enter [agree] to continue:
```

按提示输入“**agree**”，回显如下信息：

```
Succeed to configure device hinic0
```

表示配置成功。

#### 步骤5 重启服务器生效。

```
reboot
```

----结束

### 6.5.2.3 恢复默认配置

如果不再需要64队列，则需要将配置恢复为默认情况。

## 前提条件

- IN200的整体驱动包中已上传到服务器OS。  
NIC驱动包及hinicconfig工具包含在软件包中，获取方法请参见[3.1 获取软件包](#)。
- 服务器OS上已安装了IN200对应的NIC驱动包。  
安装方法请参见[3.2.1 NIC驱动维护](#)。

## 操作步骤

**步骤1** 登录服务器OS。

**步骤2** 在操作系统命令行执行如下命令进入驱动包解压后的目录，例如：/home。

```
cd /home
```

**步骤3** 执行如下命令查询当前的配置模板。

```
cd tools/linux/nic/config
```

```
./hinicconfig hinic0
```

- 若设备没有配置过模板，则回显如下信息：

```
Cannot find template in card, maybe it's empty.  
Note: the template is empty by default.
```

- 若设备配置过模板，则回显如下信息：

```
Device current configuration:  
[BASIC]  
MODENAME: S2100_SRV_NIC_3PF  
HN: 0  
PN: 2  
PM: 0x0007  
PH: 0x00000000  
INP: 7  
VINP: 4  
[VNP]  
PF00 PF01 PF02 PF03 PF04 PF05 PF06 PF07 PF08 PF09 PF10 PF11 PF12 PF13 PF14 PF15  
60 60 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
[QNP]  
PF00 PF01 PF02 PF03 PF04 PF05 PF06 PF07 PF08 PF09 PF10 PF11 PF12 PF13 PF14 PF15  
63 63 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
[VQNP]  
PF00 PF01 PF02 PF03 PF04 PF05 PF06 PF07 PF08 PF09 PF10 PF11 PF12 PF13 PF14 PF15  
03 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
[QPNP]  
PF00 PF01 PF02 PF03 PF04 PF05 PF06 PF07 PF08 PF09 PF10 PF11 PF12 PF13 PF14 PF15  
131072 131072 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
[VQPNP]  
PF00 PF01 PF02 PF03 PF04 PF05 PF06 PF07 PF08 PF09 PF10 PF11 PF12 PF13 PF14 PF15  
1024 1024 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
[CQNP]  
PF00 PF01 PF02 PF03 PF04 PF05 PF06 PF07 PF08 PF09 PF10 PF11 PF12 PF13 PF14 PF15  
131072 131072 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
[VCQNP]  
PF00 PF01 PF02 PF03 PF04 PF05 PF06 PF07 PF08 PF09 PF10 PF11 PF12 PF13 PF14 PF15  
131072 131072 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

**步骤4** 执行如下命令恢复默认配置。

```
hinicadm reset -i hinic0
```

回显如下信息：

```
Pf0 restore factory settings succeed.  
Pf1 restore factory settings succeed.
```

Pf2 restore factory settings succeed.  
Please reboot OS to take effect.

**步骤5** 重启服务器生效。

**reboot**

----结束

## 6.6 日志收集

### 6.6.1 驱动和固件日志收集

当IN200驱动或固件等发生故障时，需要收集日志信息进行故障诊断。IN200提供了一键式日志收集脚本，可以在故障发生后的第一时间进行数据收集，保证数据原始性。

#### 6.6.1.1 安装日志收集脚本

##### Linux 系统下安装日志收集脚本

一键式日志收集脚本免安装。获取日志收集脚本文件“hnic\_log\_collect.sh”请参见[3.1 获取软件包](#)。

##### VMware 系统下安装日志收集脚本

NIC驱动安装完成后，一键式日志收集脚本即自动安装完成。NIC驱动安装请参见[3.2.1.3 安装驱动](#)。

#### 6.6.1.2 运行日志收集脚本

##### Linux 系统下执行日志收集脚本

**步骤1** 登录服务器OS。

**步骤2** 参见[3.1 获取软件包](#)获取日志收集脚本文件“hnic\_log\_collect.sh”。

**步骤3** 将日志收集脚本文件上传至服务器OS的任意目录下，例如：/home。

**步骤4** 执行如下命令收集全量日志文件。

**bash hnic\_log\_collect.sh -a**

##### 说明

- 收集的全量日志文件以压缩包的形式存储在“/opt/hnic/”下，压缩包名称为“log\_[yyyy]\_[mm]\_[dd]\_[hh]\_[mm]\_[ss].tar.gz”。其中：  
[yyyy]\_[mm]\_[dd]\_[hh]\_[mm]\_[ss]为执行脚本时间，格式为“年\_月\_日\_时\_分\_秒”。例如压缩包的名称为“log\_2019\_07\_31\_07\_11\_02.tar.gz”，表示在2019年7月31日7点11分2秒生成的全量日志文件压缩包。
- 具体文件请见[表6-2](#)。

----结束

## VMware 系统下执行日志收集脚本

**步骤1** 登录服务器OS。

**步骤2** 进入日志收集脚本“hinic\_vmw\_collect.py”所在目录“/opt/huawei/”。

```
cd /opt/huawei/
```

### □ 说明

NIC驱动安装完成后，一键式信息收集脚本“hinic\_vmw\_collect.py”即自动安装完成。NIC驱动安装请参见[3.2.1.3 安装驱动](#)。

**步骤3** 执行如下命令收集全量日志文件。

```
./hinic_vmw_collect.py
```

### □ 说明

- 收集的全量日志文件以压缩包的形式存储在“/scratch/huawei/collect/”下，压缩包名称为“VMW\_huawei\_NIC\_log\_[yyyymmdd]\_[hhmmss].tgz”。其中：[yyyymmdd]\_[hhmmss]为执行脚本时间，格式为“年月日\_时分秒”。例如压缩包的名称为“VMW\_huawei\_NIC\_log\_20190731\_071113.tgz”，表示在2019年7月31日7点11分13秒生成的全量日志文件压缩包。
- 具体文件请见[表6-3](#)。

----结束

### 6.6.1.3 日志文件清单

#### Linux 系统下日志文件清单

Linux系统下日志文件清单如[表6-2](#)所示。

表 6-2 Linux 系统下日志文件清单

编号	日志文件名称	文件说明
1	*dmesg.txt	内核崩溃日志
2	[dev_id]_counter.log	微码统计（全量）
3	[dev_id]_event.log	SDK异常事件
4	[dev_id]_full_eye.log	全眼图
5	[dev_id]_hilink_count.log	统计信息
6	[dev_id]_hilink_dump.log	an_train寄存器
7	[dev_id]_hilink_param.log	Hi16参数
8	[dev_id]_hilink_port.log	端口信息
9	[dev_id]_info.log	设备形态
10	[dev_id]_mac.log	固化MAC表

编号	日志文件名称	文件说明
11	[dev_id]_mips_epc.log	微码线程信息
12	[dev_id]_mode.log	模式 ( NIC、OVS等 )
13	[dev_id]_pf_rate.log	PF限速信息
14	[dev_id]_port_autoneg.log	端口自协商协议类型
15	[dev_id]_port_fec.log	端口FEC模式信息
16	[dev_id]_port_lro.log	端口LRO设置
17	[dev_id]_port_sfp.log	端口光模块信息
18	[dev_id]_qos.log	Qos配置
19	[dev_id]_serdes.log	四点眼图
20	[dev_id]_table.log	表项信息
21	[dev_id]_temperature.log	核温、光模块温度
22	[dev_id]_version.log	驱动版本信息
23	collect_info.log	执行用户
24	cpuinfo.log	CPU信息
25	crash/	UVP内核崩溃日志
26	ethtool_info.log	网口统计信息
27	free.log	内存占用
28	ifconfig.log	网络设备信息
29	interrupts_1.log , interrupts_2.log	中断信息
30	iptables.log	防火墙状态
31	irqbalance.log	中断均衡器状态
32	logdump-[num]-[date].tar.gz	UVP内核压缩日志
33	lsb_release.log	OS发行版信息
34	lscpu.log	CPU信息
35	lspci.log	PCIe设备信息
36	meminfo.log	内存信息
37	message-[date].*	x86服务器内核压缩日志
38	messages	内核日志
39	messages.[num].gz	ARM服务器内核压缩日志

编号	日志文件名称	文件说明
40	numa_node.log	numa节点信息
41	operations.log	操作记录
42	os_info.log	Linux内核版本
43	ps.log	全格式显示进程
44	smp_affinity.log	网络设备中断绑定信息
45	snmp_1.log, snmp_2.log	协议栈统计信息
46	stat_1.log, stat_2.log	CPU统计信息
47	top.log	CPU占用率
48	ucode_flash_[dev_id]_[date].log	Flash微码日志
49	ucode_lastword_flash_[dev_id]_[date].log	微码临终遗言信息
50	ucode_ram_[dev_id]_[date].log	RAM微码日志
51	up_flash_[dev_id]_[date].log	Flash UP日志
52	up_lastword_flash_[dev_id]_[date].log	UP临终遗言信息
53	up_ram_[dev_id]_[date].log	RAM UP日志

## VMware 系统下日志文件清单

VMware系统下日志文件清单如表6-3所示。

表 6-3 VMware 系统下日志文件清单

编号	日志文件名称	文件说明
1	esxcfg-vswitch.log	系统当前所有虚拟交换机配置信息
2	esxcfg_nic.log	系统当前所有网卡物理端口信息
3	esxcfg_vmknic.log	系统当前所有vmk端口信息
4	esxi_version.log	系统当前OS版本信息
5	hinicX_counter.log	指定网卡设备的所有固件统计信息
6	hinicX_event.log	指定网卡设备的所有固件事件统计信息
7	hinicX_info.log	指定网卡设备的板卡信息

编号	日志文件名称	文件说明
8	hinicX_version.log	指定网卡设备的驱动、固件版本信息
9	hinic_info.log	系统当前所有网卡的板卡信息
10	hinic_modinfo.log	hinic驱动的内核模块信息
11	shell.log	系统当前所有的用户操作命令历史记录
12	vmkernel.log	内核的系统日志
13	vmnicX_privstats.log	指定端口的私有报文统计信息
14	vmnicX_properties.log	指定端口的设备信息
15	vmnicX_stats.log	指定端口的系统报文统计信息
16	vmkernel-log.X	vmkernel系统coredump log日志
17	vmkernel-zdump.X	vmkernel系统coredump zdump 格式文件

## 6.6.2 PXE 日志收集

当IN200 PXE出现故障时，可通过iBMC一键收集日志信息。

### 操作步骤

步骤1 登录服务器iBMC Web，详细操作请参见[iBMC用户指南](#)。

步骤2 依次选择“信息 > 信息概况 > 常用操作快捷入口 > 一键收集信息”，如图6-16所示。

图 6-16 一键式信息收集



**步骤3** 单击“一键收集信息”。

界面提示正在收集信息。

**步骤4** 收集完成后弹出“另存为”窗口，显示收集文件“dump\_info.tar.gz”，选择本地路径将其下载至本地PC机。

----结束

## 6.7 常用 BIOS 配置

### 6.7.1 在 x86 架构下常用 BIOS 配置

下面以2288H V5的BIOS为例进行说明。

#### □ 说明

操作之前，请保证服务器启动模式为“UEFI”。

**步骤1** 登录服务器远程虚拟控制台。

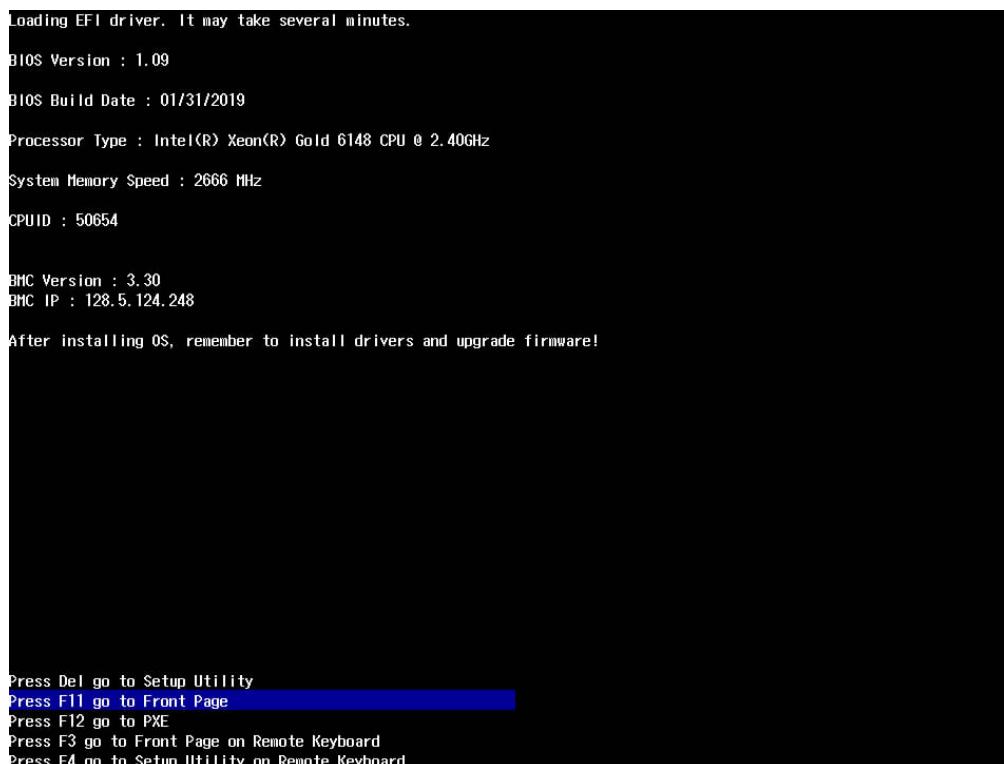
操作方法请参见[iBMC用户指南](#)。

**步骤2** 重启服务器。

**步骤3** 进入BIOS配置界面。

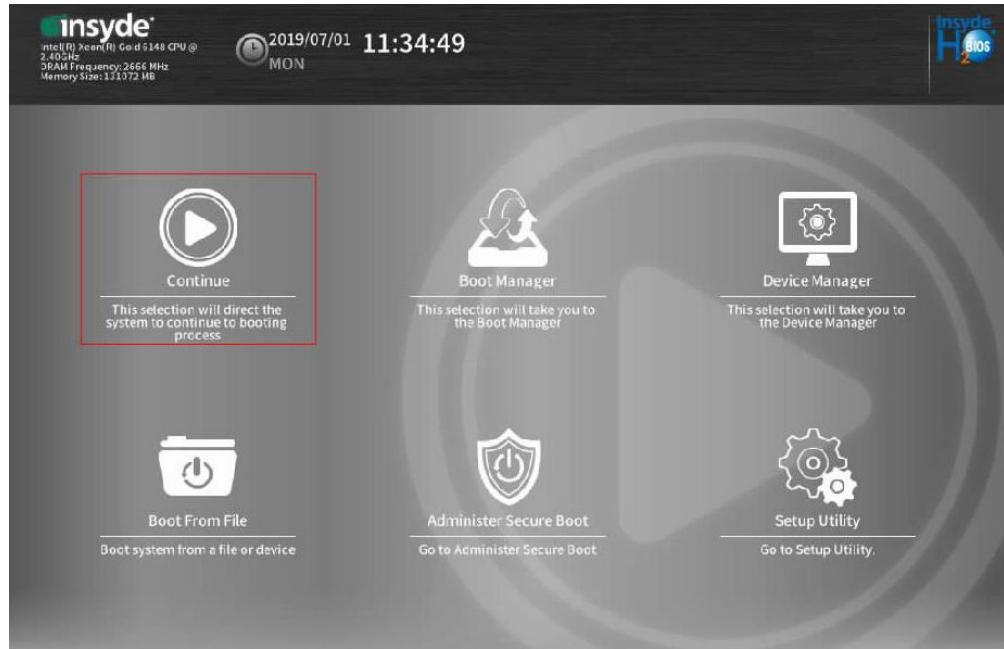
- 当BIOS的版本在V3XX及以下时，界面显示如图6-17。

图 6-17 启动信息 (1)



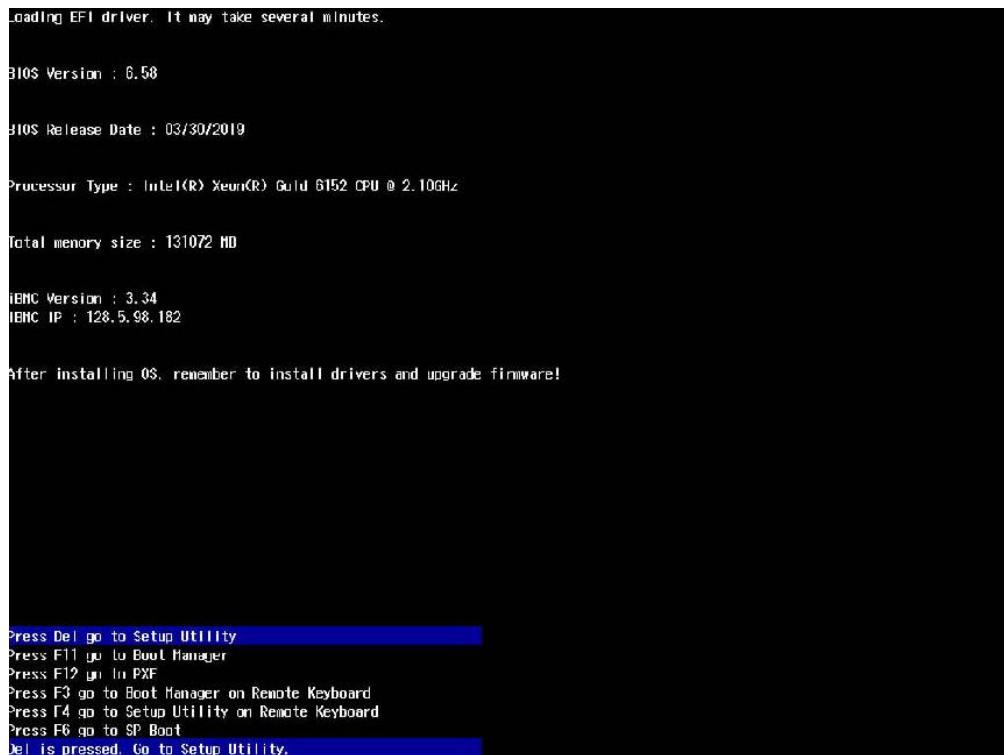
按“F11”进入“BIOS”界面，如图6-18所示，执行**步骤4**。

图 6-18 “BIOS” 界面 (1)



- 当 BIOS 的版本在 V6XX 及以上时，界面显示如图 6-19。

图 6-19 启动信息 (2)



按“Delete”进入“BIOS”界面，如图 6-20 所示，执行步骤 4。

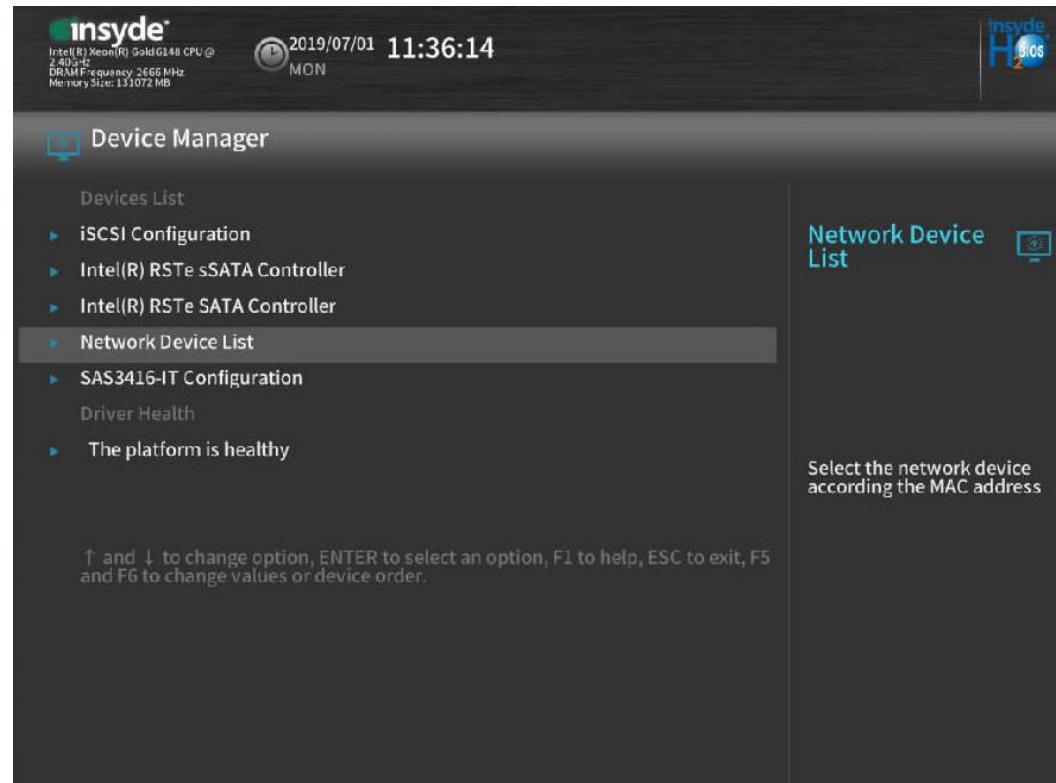
图 6-20 “BIOS”界面（2）



**步骤4** 用方向键选择“Device Manager”并按“Enter”。

打开“Device Manager”界面，如图6-21所示。

图 6-21 “Device Manager”界面



**步骤5** 使用方向键选择“Network Device List”并按“Enter”。

打开网卡列表，如图6-22所示。

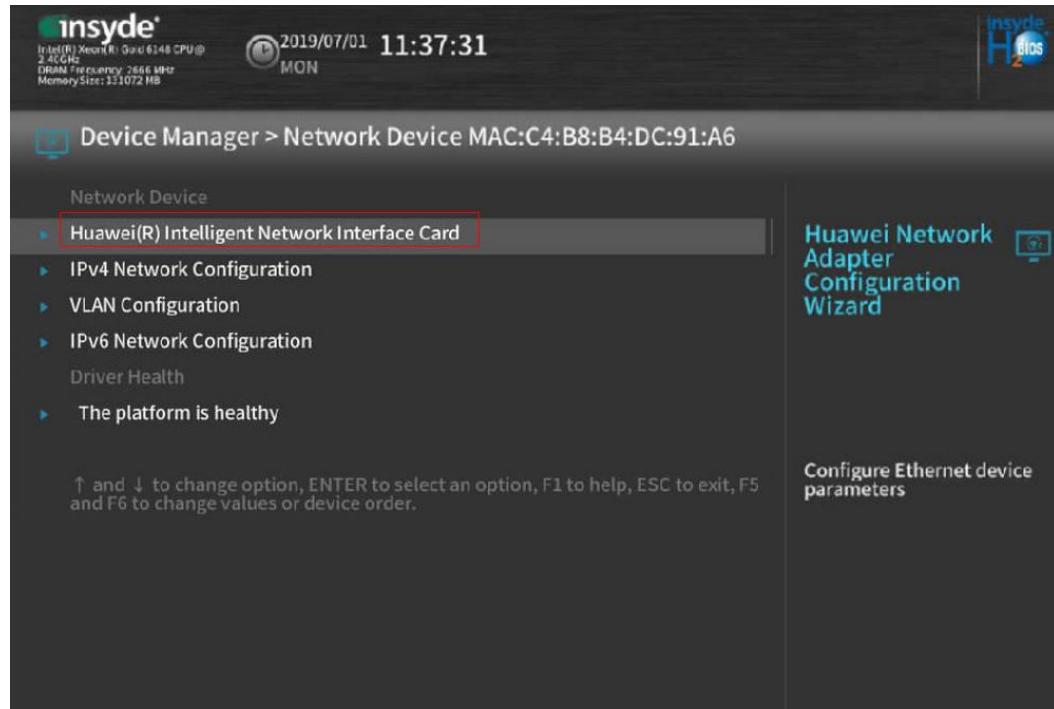
图 6-22 “Network Device List”界面



**步骤6** 按照MAC地址选择要操作的网卡，并按“Enter”。

进入网卡配置界面，如图6-23所示。

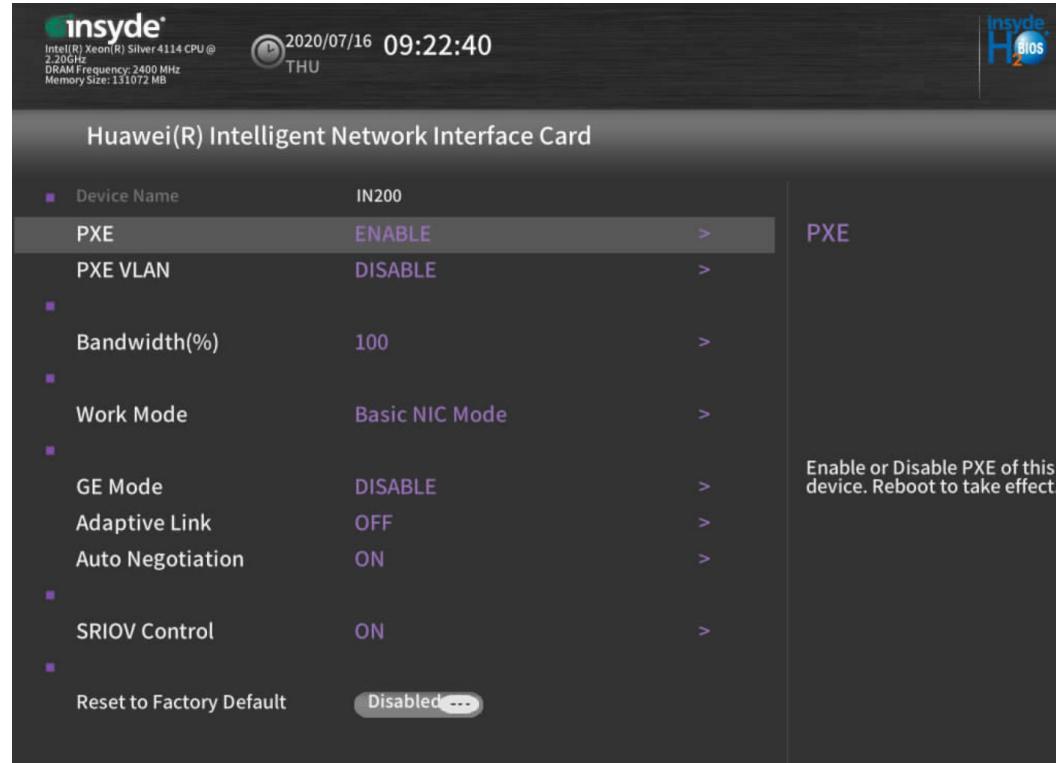
图 6-23 网卡配置界面



**步骤7** 选择“Huawei(R) Intelligent Network Interface Card”，并按“Enter”。

进入网卡控制器配置界面，如图6-24所示。

**图 6-24** “Huawei(R) Intelligent Network Interface Card” 界面



可配置的参数及其含义如表6-4所示。

**表 6-4** 配置项说明

配置项	含义	取值范围
PXE	PXE功能启用状态	<ul style="list-style-type: none"><li>● ENABLE</li><li>● DISABLE</li></ul>
PXE VLAN	VLAN功能启用状态	<ul style="list-style-type: none"><li>● ENABLE</li><li>● DISABLE</li></ul>
Bandwidth(%)	网口的速率限制百分比	1~100
Work Mode	网卡工作模式	<ul style="list-style-type: none"><li>● Basic NIC Mode</li><li>● OVS Offload Mode</li></ul>

配置项	含义	取值范围
GE Mode	<ul style="list-style-type: none"><li>使能或去使能GE模式，使能后为GE模式，仅用于调试场景。去使能后可配置具体模式，如自协商等。</li></ul> <p><b>说明</b> 配置网卡的连接模式时，port0和port1联动，port2和port3联动。例如，将port0的连接模式由enabled修改为disabled并重启生效后，port1的连接模式自动变为disabled。<ul style="list-style-type: none"><li>IN200默认去使能GE模式。</li></ul></p>	<ul style="list-style-type: none"><li>DISABLE</li><li>ENABLE</li></ul>
Adaptive Link	<p>网口的自适应模式。</p> <p><b>说明</b> Adaptive Link为OFF时，Auto Negotiation可以配置。</p>	<ul style="list-style-type: none"><li>ON</li><li>OFF</li></ul>
Auto Negotiation	<p>在网口的“GE Mode”连接模式配置为Disable情况下，网口的自协商启用状态。</p> <p><b>说明</b> Auto Negotiation为OFF时Link Speed and FEC可以配置。</p>	<ul style="list-style-type: none"><li>ON</li><li>OFF</li></ul>
Link Speed and FEC	网口的速率和FEC模式。	<ul style="list-style-type: none"><li>当网卡支持25GE和10GE的速率时，网口的速率和FEC模式有如下选择：<ul style="list-style-type: none"><li>- 25GE, NO FEC</li><li>- 25GE, BASE FEC</li><li>- 25GE, RS FEC</li><li>- 10GE, NO FEC</li><li>- 10GE, BASE FEC</li></ul></li><li>当网卡支持100GE和40GE的速率时，网口的速率和FEC模式有如下选择：<ul style="list-style-type: none"><li>- 100GE, RS FEC</li><li>- 100GE, NO FEC</li><li>- 40GE, NO FEC</li><li>- 40GE, BASE FEC</li></ul></li></ul>
SRIOV Control	配置端口的SR-IOV能力的启用状态，默认为打开。	<ul style="list-style-type: none"><li>ON</li><li>OFF</li></ul>
Reset to Factory Default	恢复出厂默认设置。	<ul style="list-style-type: none"><li>Disable</li><li>Enable</li></ul>

**步骤8** 按照实际需要进行选择和配置。

**步骤9** 保存退出。

----结束

## 6.7.2 在 ARM 架构下常用 BIOS 配置

下面以TaiShan 200（型号 2280）的BIOS为例进行说明。

### 说明

操作之前，请保证服务器启动模式为“UEFI”。

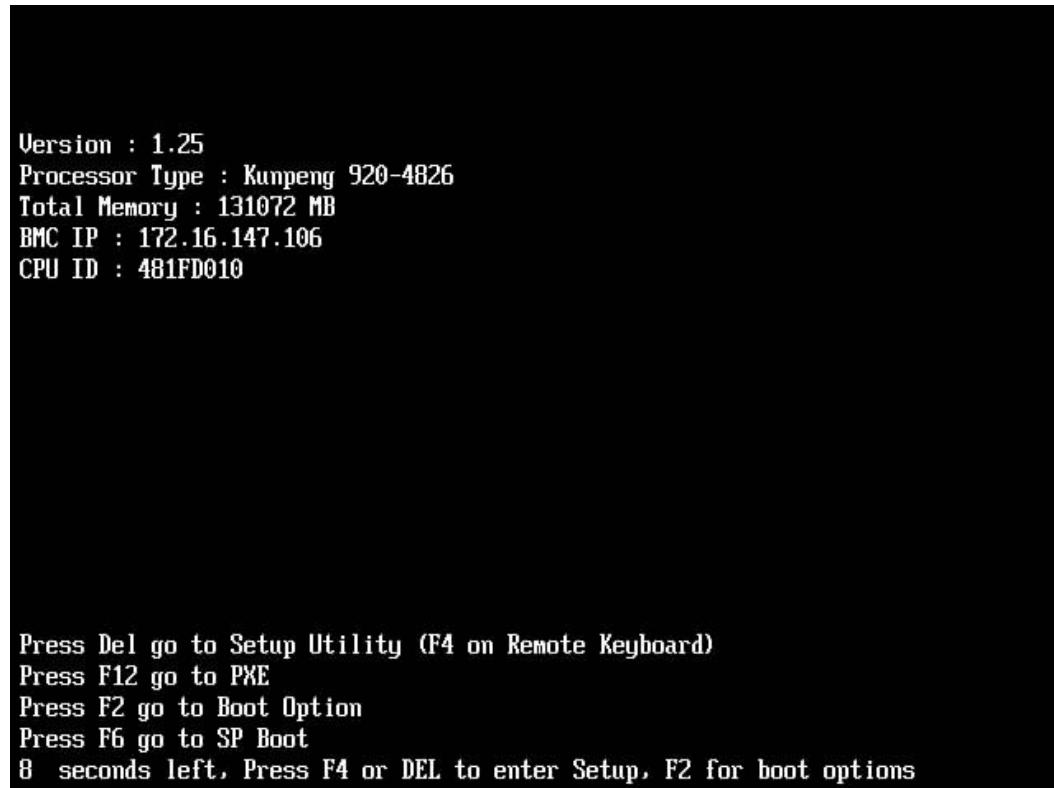
**步骤1** 登录服务器远程虚拟控制台。

操作方法请参见[iBMC用户指南](#)。

**步骤2** 重启服务器。

**步骤3** 进入BIOS配置界面。

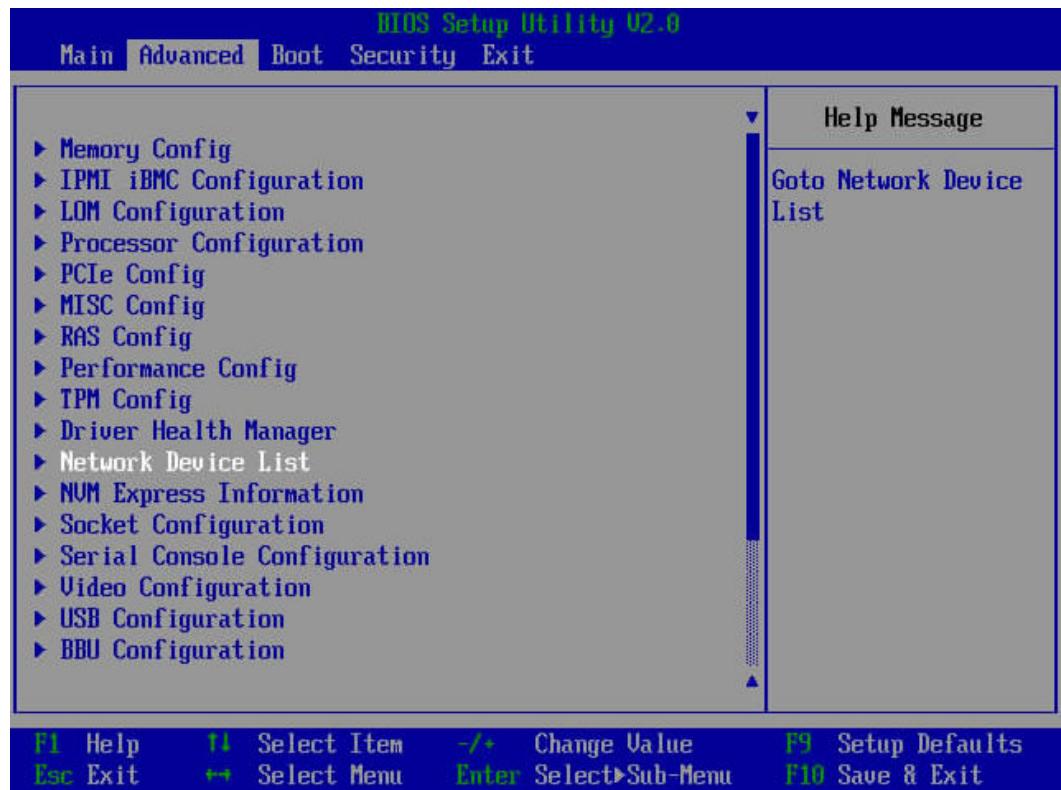
图 6-25 启动首页



按“F4”或“Delete”进入密码输入界面，输入BIOS密码后进入BIOS设置界面。

**步骤4** 用方向键选择“Advanced”页签并按“Enter”，如图6-26所示。

图 6-26 “Advanced” 界面



**步骤5** 使用方向键选择“Network Device List”并按“Enter”。

打开网卡列表，如图6-27所示。

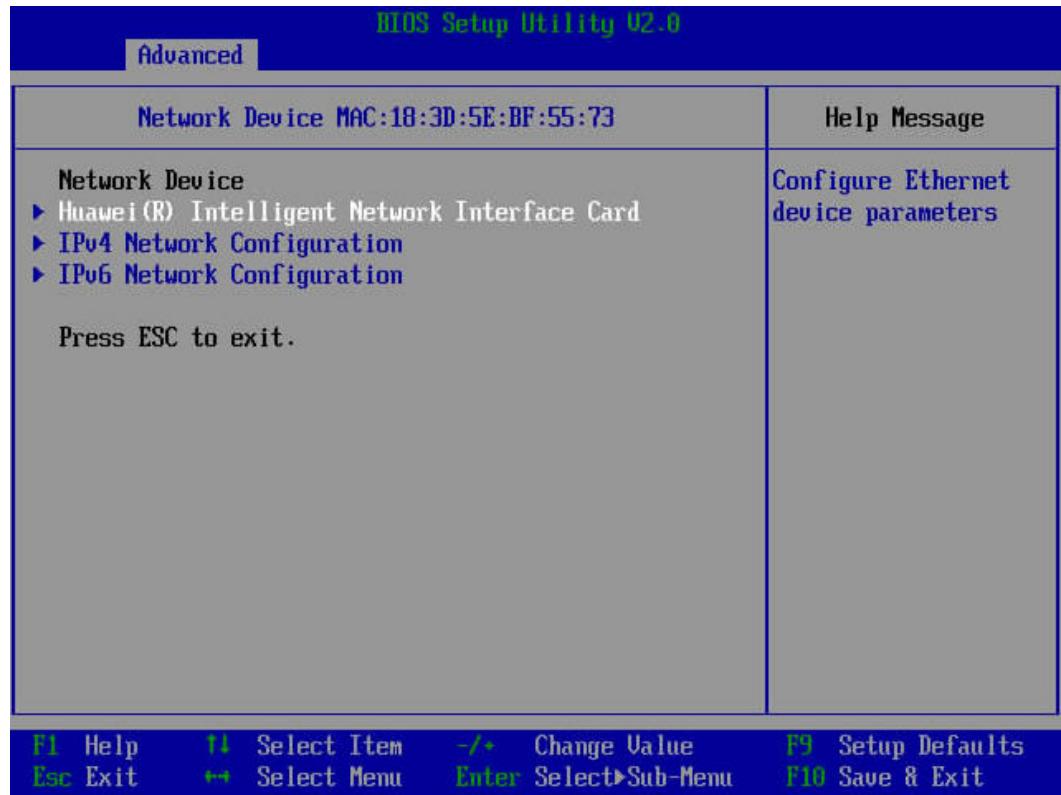
图 6-27 “Network Device List” 界面

Network Device List		Help Message
Network Device List		Network Device
▶ MAC:18:3D:5E:BF:55:71		
▶ MAC:18:3D:5E:BF:55:72		
▶ MAC:18:3D:5E:BF:55:73		
▶ MAC:18:3D:5E:BF:55:74		
▶ MAC:FA:11:47:07:13:10		
▶ MAC:FA:11:47:07:13:11		
▶ MAC:C8:A7:76:92:A9:72		
▶ MAC:C8:A7:76:92:A9:73		
▶ MAC:C8:A7:76:92:A9:74		
▶ MAC:C8:A7:76:92:A9:75		
▶ MAC:18:3D:5E:BE:CA:5B		
▶ MAC:18:3D:5E:BE:CA:5C		
▶ MAC:18:3D:5E:BE:CA:5D		
▶ MAC:18:3D:5E:BE:CA:5E		
Press ESC to exit.		
F1 Help	↓ Select Item	-/+ Change Value
Esc Exit	↔ Select Menu	Enter Select Sub-Menu
		F9 Setup Defaults
		F10 Save & Exit

步骤6 按照MAC地址选择要操作的网卡，并按“Enter”。

进入网卡配置界面，如图6-28所示。

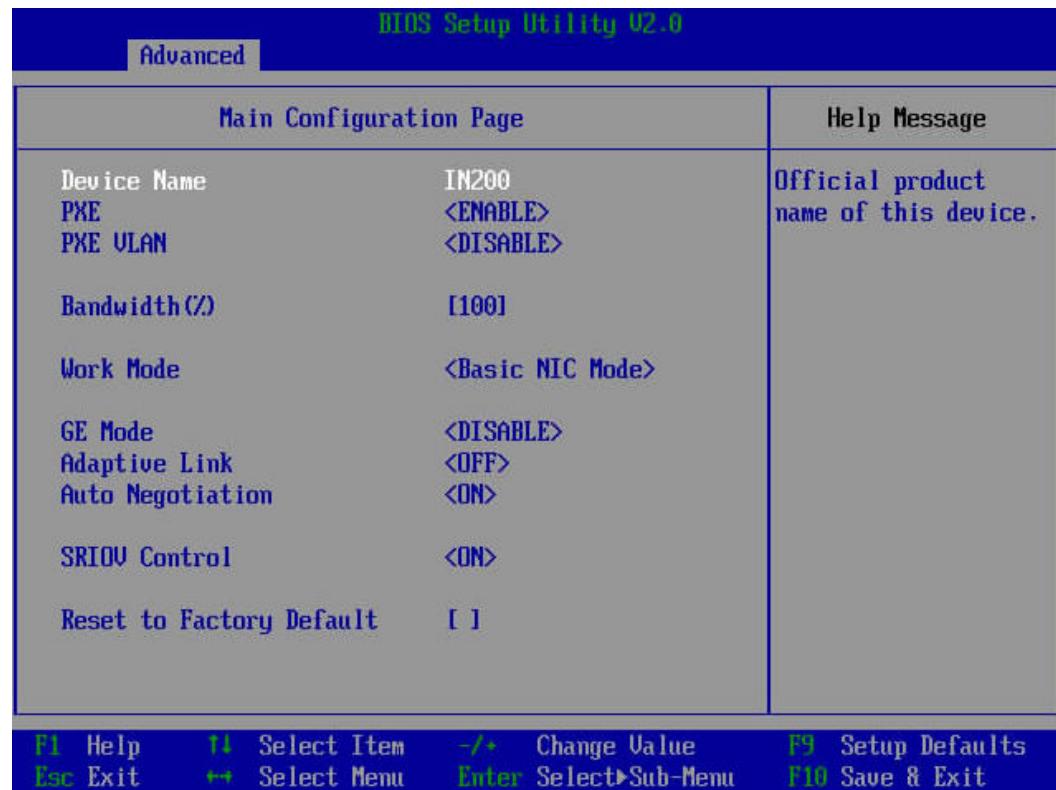
图 6-28 网卡配置界面



**步骤7** 选择“Huawei(R) Intelligent Network Interface Card”，并按“Enter”。

进入网卡控制器配置界面，如图6-29所示。

图 6-29 “Huawei (R) Intelligent Network Interface Card” 界面



可配置的参数及其含义如表6-5所示。

表 6-5 配置项说明

配置项	含义	取值范围
PXE	PXE功能启用状态	<ul style="list-style-type: none"><li>• ENABLE</li><li>• DISABLE</li></ul>
PXE VLAN	VLAN功能启用状态	<ul style="list-style-type: none"><li>• ENABLE</li><li>• DISABLE</li></ul>
Bandwidth(%)	网口的速率限制百分比	1~100
Work Mode	网卡工作模式	<ul style="list-style-type: none"><li>• Basic NIC Mode</li><li>• OVS Offload Mode</li></ul>

配置项	含义	取值范围
GE Mode	<ul style="list-style-type: none"><li>使能或去使能GE模式，使能后为GE模式，仅用于调试场景。去使能后可配置具体模式，如自协商等。</li></ul> <p><b>说明</b> 配置网卡的连接模式时，port0和port1联动，port2和port3联动。例如，将port0的连接模式由enabled修改为disabled并重启生效后，port1的连接模式自动变为disabled。<ul style="list-style-type: none"><li>IN200默认去使能GE模式。</li></ul></p>	<ul style="list-style-type: none"><li>DISABLE</li><li>ENABLE</li></ul>
Adaptive Link	<p>网口的自适应模式。</p> <p><b>说明</b> Adaptive Link为OFF时，Auto Negotiation可以配置。</p>	<ul style="list-style-type: none"><li>ON</li><li>OFF</li></ul>
Auto Negotiation	<p>在网口的“GE Mode”连接模式配置为Disable情况下，网口的自协商启用状态。</p> <p><b>说明</b> Auto Negotiation为OFF时Link Speed and FEC可以配置。</p>	<ul style="list-style-type: none"><li>ON</li><li>OFF</li></ul>
SRIOV Control	配置端口的SR-IOV能力的启用状态，默认为打开。	<ul style="list-style-type: none"><li>ON</li><li>OFF</li></ul>
Reset to Factory Default	恢复出厂默认设置。	<ul style="list-style-type: none"><li>Disable</li><li>Enable</li></ul>

**步骤8** 按照实际需要进行选择和配置。

**步骤9** 设置完成后，按“F10”。

弹出“Save configuration changes and exit?”对话框。

图 6-30 保存退出



**步骤10** 选择“Yes”，按“Enter”，保存设置。

----结束

# A FAQ

## A.1 IN200 网卡 VF 进行 FLR 时概率性出现虚拟机操作系统加载驱动失败的处理方法

### 问题现象

服务器安装Linux操作系统，且开启SR-IOV，IN200网卡多个VF（Virtual Function）分别分配给不同的Linux操作系统的虚拟机，当多个虚拟机同时复位时会概率性出现虚拟机操作系统下加载IN200网卡驱动失败问题。

### 问题原因

PCIe标准协议中规定单个VF的FLR（Function Level Reset）的完成时间在100ms以内，IN200网卡处理单个VF的FLR时间为100ms以内，满足协议要求。但当多个虚拟机同时复位时会触发多个VF同时进行FLR，而IN200网卡处理FLR的流程是串行的，会导致后面处理的VF的等待时间较长，会概率性超出100ms。主机Host的OS在等待100ms后会允许虚拟机开始启动，虚拟机启动阶段会加载IN200的VF驱动，如果在加载驱动时，该VF的FLR还未结束，则会导致驱动加载失败。

### 解决方案

- 规避措施  
逐一重启异常虚拟机。
- 解决措施  
安装IN200网卡补丁patch，实现服务器操作系统和IN200网卡固件增加VF的FLR握手机制，当前服务器操作系统下发起VF的FLR时，虚拟机操作系统将等待其所分配的VF完成FLR后继续启动操作系统，从而避免因VF未在100ms以内完成FLR导致虚拟机无法正常加载IN200网卡驱动。

# B 通过虚拟文件夹传输文件

在安装和升级驱动、升级固件、安装工具之前，都需要将对应的软件包传输到服务器OS中。

下面介绍通过虚拟远程控制台的虚拟文件夹功能传输文件的操作方法。

**步骤1** 通过远程虚拟控制台登录服务器实时桌面。

操作方法请参见[iBMC 用户指南](#)。

**步骤2** 在远程虚拟控制台的工具栏中单击。

打开虚拟设备列表，如图B-1所示。

**图 B-1** 虚拟设备菜单



**步骤3** 选择“本地文件夹”并单击之后的“浏览”。

**步骤4** 选择待传输文件所在的本地路径。

**步骤5** 单击“连接”。

**步骤6** 拷贝文件到服务器OS。

不同类型的操作系统下，“本地文件夹”表现出来的设备类型和位置不同，详细信息请参见各OS厂商官网资料。

----结束

# C 缩略语

## A

**AC** Alternating Current 交流电

## B

**BIOS** Basic Input and Output System 基本输入输出系统

## D

**DCB** Data Center Bridging 数据中心桥接

**DPDK** Data Plane Development Kit 数据平面开发套件

## E

**ETS** Enhanced Transmission Selection 扩展集群系统

## F

**FEC** Forward Error Correction 前方错误控制

**FLR** Function Level Reset 功能级复位

## I

**I<sup>2</sup>C** Inter-integrated Circuit 内部整合电路

## L

**LRO** large receive offload 大量接收负载转移

**LACP** Link Aggregation Control Protocol 链路聚合控制协议

**M**

<b>MAC</b>	Media Access Control	媒体接入控制
<b>MCTP</b>	Management Component Transport Protocol	管理组件传输协议

**N**

<b>NIC</b>	Network Interface Card	网络接口卡
<b>NVGRE</b>	Network Virtualization using Generic Routing Encapsulation	GRE的网络虚拟化

**O**

<b>OVS</b>	Open vSwitch	开放虚拟交换标准
------------	--------------	----------

**P**

<b>PCIe</b>	Peripheral Component Interconnect Express	快捷外围部件互连标准
<b>PFC</b>	Priority Flow Control	优先级流控
<b>PLDP</b>	Partner Linux Driver Program	合作伙伴Linux驱动程序计划
<b>PXE</b>	Preboot eXecution Enviroment	预启动执行环境

**R**

<b>RoCE</b>	Remote Direct Memory Access over Converged Ethernet	基于融合以太网的RDMA
<b>RSS</b>	Receive Side Scaling	接收端调节

**S**

<b>SMBus</b>	System Management Bus	系统管理总线
<b>SR-IOV</b>	Single Root I/O Virtualization	单根I/O虚拟化

**T**

<b>TC</b>	Traffic Classifier	流分类器
<b>TSO</b>	TCP Segmentation Offload	TCP分段卸载

**U**

**UEFI** Unified Extensible Firmware Interface 统一可扩展固件接口

**V**

**VLAN** Virtual Local Area Network 虚拟局域网

**VXLAN** Virtual eXtensible Local Area Network 虚拟扩展局域网