

排除故障NCS6K卡德引导失败

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[故障排除](#)

[线路卡引导装入正如所料](#)

[结构卡无法启动由于驱动程序错误](#)

[卡德在POWEROFF或现状停留](#)

[卡德在POWER ON状态停留](#)

[卡德在SW INACTIVE状态停留](#)

[方案1. SW_EVENT_FAILURE : shelf_mgr Syslog报告的SW_EVENT_ADMIN_VM_FAILURE](#)

[在故障状态滞留的方案2. LC, 最后事件 : HW_EVENT_FAILURE](#)

[fail_code=LC_POWER_MAIN_FAULT](#)

[卡德在UNKNOWN状态停留](#)

[请使用RCONSOLE工具](#)

[命令列表收集, 在您开TAC案例前](#)

简介

本文描述如何排除故障网络收敛系统6000 (NCS6K)线卡引导失败。另外，它也提供可以收集可帮助TAC进行一次详细的调查数据的概述。

先决条件

要求

Cisco建议您有XR命令行界面(CLI)基础知识。

使用的组件

使用XR发行版本5.0.1, 5.2.1, 5.2.3和5.2.4, 本文创建。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始(默认)配置。如果您使用的是真实网络, 请确保您已经了解所有命令的潜在影响。

背景信息

如果路由处理器(RP)、结构卡(FC)或线卡(LC)是不启动和陷在引导程序阶段, 第一步应该将开始收集show tech ctrace从Syadmin虚拟机。此show tech在当前系统状态提供不同的组件之间的信息和

交互作用。然而，有机会亦称Sysadmin卡尔瓦多斯不在卡，并且show tech ctrace不能收集受影响的卡的信息。因为卡通过安全壳SSH，不是可及的这会发生。对于这样案件，rconsole步骤将是必要为了知道卡获得为什么在启动程序时停留。

Note:此文件通常是相当大的(500MB-1GB)和存储在Sysadmin VM。为了从方框解压缩它，必须复制它到XR VM (说明提供了以后在同一个文档)。

故障排除

检查每个卡状态在Sysadmin VM的并且发现其当前状态。注意特别注意H/w和S/w状态。注意显示S/w状态的卡，因为N/A是CPU较少卡(FC卡、风扇盘等等)哪些根本是由RP CPU控制的。卡在操作状态是CPU基于卡并且有加载的软件。

```
sysadmin-vm:0_RP0# show platform
```

Location	Card Type	HW State	SW State	Config State
0/0	PROTO-CXP-1XPITA	OPERATIONAL	OPERATIONAL	NSHUT
0/2	PROTO-CXP-2XPITA	POWERED_OFF	SW_INACTIVE	NSHUT
0/3	NC6-10X100G-M-K	OPERATIONAL	OPERATIONAL	NSHUT
0/RP0	NC6-RP	OPERATIONAL	OPERATIONAL	NSHUT
0/RP1	NC6-RP	POWERED_ON	SW_INACTIVE	NSHUT
0/FC0	NC6-FC	POWERED_ON	N/A	NSHUT
0/FC1	NC6-FC-MC	POWERED_ON	N/A	NSHUT
0/FC2	UNKNOWN	FAILED	N/A	NSHUT
0/FC3	NC6-FC	POWERED_ON	N/A	NSHUT
0/FC4	NC6-FC-B2B	POWERED_ON	N/A	NSHUT
0/FC5	NC6-FC	OPERATIONAL	N/A	NSHUT
0/FT0	NC6-FANTRAY	OPERATIONAL	N/A	NSHUT
0/FT1	NC6-FANTRAY	OPERATIONAL	N/A	NSHUT
0/PT0	NCS-AC-PWRTRAY	OPERATIONAL	N/A	NSHUT
0/PT1	NCS-AC-PWRTRAY	OPERATIONAL	N/A	NSHUT
0/PT2	NCS-AC-PWRTRAY	OPERATIONAL	N/A	NSHUT

下一步是对检查卡片芯片控制器(CCC)库存命令并且确认卡状态。

```
sysadmin-vm:0_RP0# show controller ccc inventory summary
```

CCC Inventory Summary :

Location	Card Type	BP ID	Serial Number	HW Ver	Card State
0/RP0	NC6-RP (master)	0	SAD15270129	0.1	CARD_READY
0/RP1	NC6-RP (slave)	1	SAD1527012P	0.1	CARD_READY
0/FC0	NC6-FC	8	SAD1618002F	0.2	WAIT_DEV_INIT
0/FC1	NC6-FC	9	SAD153901ZT	0.2	WAIT_DEV_INIT
0/FC4	NC6-FC	12	SAL1803KQEY	1.0	PON_POWERING_UP
0/FC5	NC6-FC	13	SAD16180043	0.2	WAIT_DEV_INIT
0/0	NC6-10X100G-M-K	16	SAL1650UCN9	0.4	PXE_BOOTING
0/4	NC6-10X100G-M-K	20	SAD154502XU	0.1	CARD_READY

这是能发生的不同的方案。并且，列出此处是预期的输出和下故障排除步骤。

线路卡引导装入正如所料

```
sysadmin-vm:F0_SC0# show platform detail location 0/0
```

```
Platform Information for 0/0
PID : NC6-10X100G-M-P
Description : "NCS 6000 10x100G Multi-Service CXP"
VID/SN : V01
HW Oper State : OPERATIONAL
SW Oper State : OPERATIONAL
Configuration : "NSHUT RST"
HW Version : 1.0
Last Event : HW_EVENT_OK
Last Event Reason : "Initial discovered state:BOOTED (card ok)"
```

最后事件和为时事件原因输出显示卡是好的。执行显示history命令的重新启动为了验证，如果此特定卡以前有问题，并且，如果是，什么是问题。

```
sysadmin-vm:F0_SC0# show reboot-history card location 0/0
```

```
Card Reboot History for 0/0
0
Timestamp "Fri Oct 2 15:15:26 2015"
Reason Code 7
Reason "Install Activate System Reload"
Src Location ""
Src Name INSTALL
Timestamp "Tue Sep 8 18:56:29 2015"
Reason Code 7
Reason "ADMIN CLI RELOAD ROUTER GRACEFUL"
Src Location ""
Src Name "CONFD USER"
```

Aborted: by user

线路卡0/0是可操作的，并且最后重新加载的原因是根本含义的安装软件维护更新(SMU)安装或软件升级完成。这预计，并且因而没有与此卡的问题。

结构卡无法启动由于驱动程序错误

```
sysadmin-vm:0_RP0# show platform
```

Location	Card Type	HW State	SW State	Config State
0/FC2	UNKNOWN	FAILED	N/A	NSHUT <-

```
sysadmin-vm:F0_SC0# show reboot-history card location 0/FC2
```

```
Card Reboot History for 0/FC2
Timestamp "Thu Oct 9 12:10:22 2014"
Reason Code 15 <--
"Board reload as devices not up on Fabric Card"
Src Location 0/FC0
Src Name FAM_AGENT_CALV_DRIVER_SFE 8
```

在本例中，因为sfe_driver未适当地，初始化FC卡没有启动。

执行少量更多命令为了查看重置历史记录从CCC观点。重新启动历史记录CLI可以与CCC驱动程序的重置历史记录CLI一道用于确定卡重新加载来源和原因。

能有两个选项：

内置-请使用此选项寻找信息，如果卡通过热重置(CPU重新加载)。

Onchip -请使用此选项寻找信息，如果卡通过冷重置(卡德完全地重新加载的-硬重置)。

例如：

```
sysadmin-vm:F0_SC0# show controller ccc reset-history on
Possible completions:
  onboard  CCC Reset history in onboard EEPROM detail information
  onchip   On-chip reset history entries since last CCC Cold Reset
```

```
sysadmin-vm:F0_SC0# show controller ccc reset-history onchip location 0/0
```

```
*****
***   On Chip Reset History for location 0/0   ***
*****
TimeofDay      : Tue Oct 20 17:17:40 2015
Uptime         : 18 days 02:01:59 <--
Resets         : 2

      Reset          Reset          Reset
idx Source          Command          Time
-----
0  ColdRst          AssrtHR          2015/10/02 15:15:43
1  ColdRst          DeAssrtHR        2015/10/02 15:15:50 --> List reset source as "ColdRst"
```

```
sysadmin-vm:F0_SC0# show controller ccc reset-history onboard location 0/0
```

```
*****
***   On Board Reset History for location 0/0   ***
*****
Scratch EEPROM Magic : PON
Scratch EEPROM Version : 0x00014000
Reset History Magic   : HIST
Number of Resets      : 102 <---

      Reset          Reset          Reset
idx Source          Command          Time
-----
0  ColdRst          AssrtHR          1970/01/01 0:00:00 <-- ColdRst
1  ColdRst          DeAssrtHR        1970/01/01 0:00:06
2  ColdRst          AssrtHR          1970/01/01 0:00:00
3  ColdRst          DeAssrtHR        1970/01/01 0:00:06
4  WarmRst          AssrtHR          1970/05/03 7:21:55 <-- WarmRst
<output omitted>
```

ColdRst can either be initiated by Software or Hardware. WarmRst, is only initiated through Software. Other reset sources can be HRESET_L, SRESET_L, Wtchdog, SW_assgn or plain Rsrvd.

另外，有每被执行的重置操作两个唯一条目。AssrtHR操作和DeAssrtHR操作。这暗示复位信号主张然后DE主张，因此，芯片将完成重置。

注意到这些操作中的每一的时间戳。此CLI可以与CCC库存状态CLI一起确定，当重置发生了，并且卡多久是上上下下。

其次，检查CCC阶段卡在其重新加载进程时经历了。列出得不同的状态示例此处：

从CCC观点正确地启动的卡德：

```
sysadmin-vm:F0_SC0# show controller ccc event-history brief location 0/0
```

```
CCC Card Event History for: 0/0
```

```
Card Event History as seen by Master (0/RP1)
```

```
Current State: CARD_READY
```

DATE	TIME (UTC)	STATE	EVENT
10/02	15:16:55.234	WAIT_BOOT_IMAGE	ev_boot_ssd_image
10/02	15:16:54.233	BIOS_STARTED	if_wait_ssd_image_booting
10/02	15:16:54.233	CPU_READY	if_bios_started
10/02	15:16:54.231	OIR_INSERT_NOTIF	if_cpu_is_ready
10/02	15:16:54.217	CCC_DRIVER_INIT	if_oir_insert_notif_not_done
10/02	15:16:54.195	PON_POWERED_ON	to_ccc_driver_init
10/02	15:16:54.195	CHECK_CCC_STATUS	if_pon_powered_on
10/02	15:16:54.194	READ_IDPROM	ev_idprom_available
10/02	15:16:53.942	GET_CCC_INFO	ev_get_ccc_info_done
10/02	15:16:53.723	WAIT_ETH_READY	ev_eth_available
10/02	15:16:52.560	CHECK_UBLAZE_BOOT	ev_ublaze_boot_ok
10/02	15:16:52.539	WAIT_CCC_READY	ev_ccc_ready
10/02	15:16:52.537	IDLE	ev_presence_scan

当前卡德PXE_BOOTING阶段的：

```
sysadmin-vm:0_RP0# show controller ccc event-history brief location 0/3
```

```
CCC Card Event History for: 0/3
```

```
Current State: PXE_BOOTING
```

DATE	TIME (UTC)	STATE	EVENT
08/07	19:50:40.607	BIOS_STARTED	if_internal_pxe_booting
08/07	19:50:40.607	WAIT_BIOS_START	ev_bios_started
08/07	19:50:18.605	CPU_READY	if_bios_not_started
08/07	19:50:18.595	CCC_DRIVER_INIT	if_cpu_is_ready
08/07	19:50:18.568	PON_POWERED_ON	to_ccc_driver_init
08/07	19:50:18.568	CHECK_CCC_STATUS	if_pon_powered_on
08/07	19:50:18.567	GET_CCC_INFO	ev_get_ccc_info_done
08/07	19:50:18.550	WAIT_ETH_READY	ev_eth_ready
08/07	19:50:18.550	CHECK_UBLAZE_BOOT	ev_ublaze_boot_ok
08/07	19:50:18.517	PON_UP_WARM	ev_ccc_reset_done
08/07	19:50:12.627	PON_DOWN_WARM	ev_pon_up_warm
08/07	19:50:08.239	PON_DOWN_WARM	ev_warm_reset_req_ignored
08/07	19:50:07.239	PON_DOWN_WARM	ev_warm_reset_req_ignored
08/07	19:50:06.239	PON_DOWN_WARM	ev_warm_reset_req_ignored
08/07	19:50:05.239	PON_DOWN_WARM	ev_warm_reset_req_ignored
08/07	19:50:04.238	PON_DOWN_WARM	ev_warm_reset_req_ignored

无法的卡德启动由于切削卡住在GET_CCC_INFO：

```
sysadmin-vm:0_RP0# show controller ccc event-history brief location 3/6
```

CCC Card Event History for: 3/6

Card Event History as seen by Master (3/RP0)

Current State: **GET_CCC_INFO**

DATE	TIME (UTC)	STATE	EVENT
10/26	23:43:04.559	UBLAZE_NOT_READY	ev_timer_expired
10/26	23:42:34.559	CHECK_UBLAZE_BOOT	ev_timer_expired
10/26	23:42:24.528	WAIT_CCC_READY	ev_ccc_ready
10/26	23:42:21.516	RECOVERY_RESET	ev_timer_expired
10/26	23:42:03.516	CHECK_UBLAZE_BOOT	ev_ublaze_pre_boot_failed
10/26	23:41:52.480	WAIT_CCC_READY	ev_ccc_ready
10/26	23:41:49.468	RECOVERY_RESET	ev_timer_expired
10/26	23:41:32.467	WAIT_CCC_READY	ev_no_fpga_ok_signal
10/26	23:41:29.456	RECOVERY_RESET	ev_timer_expired
10/26	23:41:13.455	WAIT_CCC_READY	ev_no_fpga_ok_signal
10/26	23:41:10.444	RECOVERY_RESET	ev_timer_expired
10/26	23:40:55.444	CHECK_UBLAZE_BOOT	ev_ublaze_pre_boot_failed
10/26	23:40:55.439	WAIT_CCC_READY	ev_ccc_ready
10/26	23:40:52.320	IDLE	ev_presence_scan

无法的卡德启动由于POWER_UP_FAILED状态：

```
sysadmin-vm:0_RP0# show controller ccc event-history brief location 0/2
```

CCC Card Event History for: 0/2

Current State: **POWER_UP_FAILED**

DATE	TIME (UTC)	STATE	EVENT
08/05	14:55:17.449	POWER_UP_FAILED	ev_wdog_timeout
08/05	14:45:31.265	CCC_DRIVER_INIT	if_pwr_up_failed
08/05	14:45:31.260	CHECK_CCC_STATUS	if_pwr_up_failed_again
08/05	14:45:31.258	GET_CCC_INFO	ev_get_ccc_info_done
08/05	14:45:31.223	WAIT_ETH_READY	ev_eth_ready
08/05	14:45:31.157	CHECK_UBLAZE_BOOT	ev_ublaze_boot_ok
08/05	14:45:31.124	PON_UP_WARM	ev_ccc_reset_done
08/05	14:45:17.489	CCC_IN_RESET	ev_pon_up_warm
08/05	14:45:08.921	POWER_UP_FAILED	ev_pon_down_warm
08/05	14:35:07.152	POWER_UP_FAILED	ev_wdog_timeout
08/05	14:25:20.946	CCC_DRIVER_INIT	if_pwr_up_failed
08/05	14:25:20.941	CHECK_CCC_STATUS	if_pwr_up_failed_again
08/05	14:25:20.939	GET_CCC_INFO	ev_get_ccc_info_done
08/05	14:25:20.923	WAIT_ETH_READY	ev_eth_ready
08/05	14:25:20.887	CHECK_UBLAZE_BOOT	ev_ublaze_boot_ok
08/05	14:25:20.830	PON_UP_WARM	ev_ccc_reset_done

Aborted: by user

如果以简要选项使用此命令，不给与问题的根本原因涉及的完整数据。对于该信息请用详细信息替换简要关键字。

Note:这是最重要的CLI，当排除故障卡时不启动在CCC级别上。

着重疏忽的事件Desc和原因获得在失败的一更加好的说明。

```
sysadmin-vm:F0_SC0# show controller ccc event-history detail location 0/0
```

```
CCC Card Event History for: 0/0
```

```
Card Event History as seen by Master (0/RP1)
```

```
Event buffer info:
```

```
Total number of events recorded: 13
```

```
Number of events available for display: 13
```

```
Current State: CARD_READY
```

```
EVENT #: 12 (record index = 12)
```

```
TIMESTAMP: 2015/10/02 15:16:55.234814 UTC
```

```
STATE: WAIT_BOOT_IMAGE
```

```
EVENT: ev_boot_ssd_image
```

```
EVENT DESC: SSD image is booting
```

```
EVENT #: 11 (record index = 11)
```

```
TIMESTAMP: 2015/10/02 15:16:54.233898 UTC
```

```
STATE: BIOS_STARTED
```

```
EVENT: if_wait_ssd_image_booting
```

```
EVENT #: 10 (record index = 10)
```

```
TIMESTAMP: 2015/10/02 15:16:54.233855 UTC
```

```
STATE: CPU_READY
```

```
EVENT: if_bios_started
```

```
EVENT #: 9 (record index = 9)
```

```
TIMESTAMP: 2015/10/02 15:16:54.231426 UTC
```

```
STATE: OIR_INSERT_NOTIF
```

```
EVENT: if_cpu_is_ready
```

```
EVENT #: 8 (record index = 8)
```

```
TIMESTAMP: 2015/10/02 15:16:54.217351 UTC
```

```
STATE: CCC_DRIVER_INIT
```

```
EVENT: if_oir_insert_notif_not_done
```

```
EVENT #: 7 (record index = 7)
```

```
TIMESTAMP: 2015/10/02 15:16:54.195808 UTC
```

```
STATE: PON_POWERED_ON
```

```
EVENT: to_ccc_driver_init
```

```
EVENT #: 6 (record index = 6)
```

```
TIMESTAMP: 2015/10/02 15:16:54.195786 UTC
```

```
STATE: CHECK_CCC_STATUS
```

这是不同的方案示例输出。

没有启动由于电源问题和被陷在的POWER_UP_FAILED的卡德：

注意ERROR_INFO获得关于失败的详细信息。

```
sysadmin-vm:0_RP0# show controller ccc event-history detail location 0/2
```

```
CCC Card Event History for: 0/2
```

```
Event buffer info:
```

```
Total number of events recorded: 692
```

```
Number of events available for display: 255
```

Current State: **POWER_UP_FAILED**

EVENT #: 691 (record index = 179)
TIMESTAMP: 2014/08/05 14:55:17.449979 UTC
STATE: POWER_UP_FAILED
EVENT: ev_wdog_timeout
EVENT DESC: CCC watchdog timeout event
ERROR INFO: wdog__0 SysAdmin VM Watchdog stage1:0

<output omitted>

没有启动和被陷在的CCC_NOT_READY的卡德(芯片问题) :

sysadmin-vm:0_RP0# **show controller ccc event-history detail location 0/FC2**

CCC Card Event History for: 0/FC2

Event buffer info:
Total number of events recorded: 2
Number of events available for display: 2

Current State: **CCC_NOT_READY**

EVENT #: 1 (record index = 1)
TIMESTAMP: 2014/08/04 14:10:49.891845 UTC
STATE: WAIT_CCC_READY
EVENT: ev_ccc_ready_timeout
EVENT DESC: Timeout waiting for CCC to be ready
ERROR INFO: CCC READY Timeout - CLOCK_OK signal not being asserted (I/O Expander port0=0xf0, port1=0xff) 8

<output omitted>

有需要删除/再插入卡的实例。为此，CCC组件给的货架跟踪卡插入/删除的提供OIR历史记录。注释关键字从使用。这将提供关于待机RP的信息。

sysadmin-vm:0_RP0# **show controller ccc oir-history rack 0**

Cards OIR History of rack: 0

OIR Events as seen by Master (0/RP0)- View from the Active RP

DATE	TIME (UTC)	EVENT	LOC	CARD TYPE	SERIAL NO
10/09	16:59:14.280	INSERTED	0/0	NC6-10X100G-M-K	SAL1650UCN9
10/09	16:58:49.064	REMOVED	0/0	NC6-10X100G-M-K	SAL1650UCN9

<output omitted>

sysadmin-vm:0_RP1# **show controller ccc slave oir-history rack 0**

Cards OIR History of rack: 0

OIR Events as seen by Slave (0/RP1)- <-- View from the standby RP

DATE	TIME (UTC)	EVENT	LOC	CARD TYPE	SERIAL NO
11/06	05:54:31.374	DISCOVERED	0/2	NC6-10X100G-M-K	SAD161300XK
11/06	05:53:37.442	DISCOVERED	0/6	NC6-10X100G-M-K	SAL1649TN46

<output omitted>

信息必须是相同的从两RP观点。

使用命令的此组合帮助确定根本原因结构卡为什么无法启动。

卡德在POWEROFF或现状停留

如果卡在POWEROFF/PRESENT状态被滞留，是很可能通过多重置并且由shelf_mgr是关闭电源。

发出这些命令为了确定问题的根本原因：

1. 收集从Sysadmin VM的show tech ctrace
2. 显示平台详细信息位置<>
3. 显示重新启动历史记录卡位置<> (请发现多少次通过重置过程)
4. show controller ccc事件历史记录详细信息位置<>
5. show controller ccc重置历史记录内位置位置<>

卡可以重置与使用hw-module reset命令，并且启动过程可以观察与使用rconsole步骤解释的以后在本文。

```
sysadmin-vm:0_RP1# show controller ccc slave oir-history rack 0
```

Cards OIR History of rack: 0

OIR Events as seen by Slave (0/RP1)- <-- View from the standby RP

DATE	TIME (UTC)	EVENT	LOC	CARD TYPE	SERIAL NO
11/06	05:54:31.374	DISCOVERED	0/2	NC6-10X100G-M-K	SAD161300XK
11/06	05:53:37.442	DISCOVERED	0/6	NC6-10X100G-M-K	SAL1649TN46

<output omitted>

卡德在POWER_ON状态停留

如果卡是卡住的POWERED_ON，意味着CCC打开是需要的为了其他驱动程序能开始他们的工作的基本电源区域。是各自驱动程序的责任迁移卡向操作状态。

亦称，在检测并且初始化在该卡ASIC后的所有其内部设备SFE驱动程序迁移FCs向操作状态。

ESD驱动程序移动SC-SW卡和Scapa的LCs向操作状态(几乎立即，什么都检查/不初始化不同于SFE驱动程序)。

如果卡德在POWERED_ON状态被滞留，意味着一个上述驱动程序有麻烦迁移卡向操作状态。问题在CPU少的卡经常被看到。例如：结构卡或SC的交换机卡(SC-SW)。

第一步将检查ccc事件历史记录命令：

```
sysadmin-vm:F0_SC0# show controller ccc event-history detail location 0/0
```

CCC Card Event History for: 0/0

Card Event History as seen by Master (0/RP1)

Event buffer info:

Total number of events recorded: 13
Number of events available for display: 13
<output ommited>

EVENT #: 7 (record index = 7)
TIMESTAMP: 2015/10/02 15:16:54.195808 UTC
STATE: PON_POWERED_ON
EVENT: to_ccc_driver_init

其次，请验证需要的基本电源区域：

```
sysadmin-vm:0_RP0# show controller ccc register location 0/RP0 offset 0x4c
```

```
Register      Register
Address       Value
-----
0x4C          0x3          - zones 0 and 1 OK
```

```
sysadmin-vm:0_RP0# show controller ccc register location 0/RP0 offset 0x50
```

```
Register      Register
Address       Value
-----
0x50          0x3          - zone 0 and 1 Enabled
```

```
sysadmin-vm:0_RP0# show controller ccc power detail location 0/RP0
```

Power detail : Zone information for 0/RP0:

```
-----
| Power Zone | Power Status | Power Contrl | Power Fault |
-----
| 0          | OK          | SET          | --          | - Power Status OK
| 1          | OK          | SET          | --          | - Power Status OK
```

```
sysadmin-vm:F0_SC0# show controller ccc i2c-dev ioexpander location 0/0
```

CCC IO Expander information for location: 0/0

Port 0: 0x3e

Port Bit I/O Val Bit Name

```
-----
P0 0 0 0 Power Cycle
P0 1 I 1 FPGA OK
P0 2 I 1 uBlaze OK
P0 3 I 1 Clock OK
P0 4 I 1 Core Volt OK
P0 5 I 1 OTH Volt0 OK
P0 6 I 0 OTH Volt1 NOT OK
P0 7 I 0 OTH Volt2 NOT OK
```

Port 1: 0x3

Port Bit I/O Val Bit Name

```
-----
P1 0 I 1 FPGA INIT OK
```

如果此验证不导致根本原因那么下一步是打开TAC服务请求。

卡德在SW_INACTIVE状态停留

```
sysadmin-vm:0_RP0# show platform
```

Location	Card Type	HW State	SW State	Config State
0/1	P-L-10X100G-F-P	POWERED_OFF	SW_INACTIVE	SHUT
0/RP0	P-L-RP	OPERATIONAL	OPERATIONAL	NSHUT
0/RP1	P-L-RP	OPERATIONAL	OPERATIONAL	NSHUT
0/FC0	P-L-FC-S	OPERATIONAL	N/A	NSHUT
0/FC1	P-L-FC-S	OPERATIONAL	N/A	NSHUT
0/FT0	PANINI-SIM-FT	OPERATIONAL	N/A	NSHUT
0/FT1	PANINI-SIM-FT	OPERATIONAL	N/A	NSHUT

可能的失败原因是：

- 主机OS不启动由于SSD访问问题
- 启动卡住由于HW问题的主机OS
- 失败的Sysadmin VM获得产生
- 控制以太网连接问题
- 卡德MAC/IP没被编程的由于SW失败
- 没获得的以太网交换机正确地编程由CCC接通电源口译员
- ESD颜色交换机镜像在CCC SPI闪存失败获得编程

方案1. SW_EVENT_FAILURE : shelf_mgr Syslog报告的SW_EVENT_ADMIN_VM_FAILURE

```
sysadmin-vm:0_RP0# show platform
```

Location	Card Type	HW State	SW State	Config State
0/1	PROTO-CXP-2XPITA	OPERATIONAL	OPERATIONAL	NSHUT
0/RP0	NC6-RP	OPERATIONAL	OPERATIONAL	NSHUT
0/RP1	NC6-RP	OPERATIONAL	SW_INACTIVE	NSHUT
0/FC0	NC6-FC-MC	OPERATIONAL	N/A	NSHUT
0/CI0	P-L-CRFT	OPERATIONAL	N/A	NSHUT
0/FT0	P-L-FANTRAY	OPERATIONAL	N/A	NSHUT
0/FT1	P-L-FANTRAY	OPERATIONAL	N/A	NSHUT

可能有几个不同的原因至于RP1为什么不启动。发现问题的简便的方法是对在RP的rconsole，并且检查注册(是本文底部)的参考的rconsole步骤。

在故障状态滞留的方案2. LC，最后事件：HW_EVENT_FAILURE fail_code=LC_POWER_MAIN_FAULT

保证HW状态显示失败，并且SW状态显示SW_INACTIVE：

```
sysadmin-vm:0_RP0# show platform location 0/1
```

Location	Card Type	HW State	SW State	Config State
----------	-----------	----------	----------	--------------

0/1 NC6-60X10GE-M-S FAILED SW_INACTIVE NSHUT

执行此命令和检查最后原因：

```
sysadmin-vm:0_RP0# show platform detail location 0/1
```

Platform Information for 0/1

```
PID : NC6-60X10GE-M-S
Description : "NCS 6000 60x10G Multi-Service SFP+"
VID/SN : V01
HW Oper State : FAILED
SW Oper State : SW_INACTIVE
Configuration : "NSHUT RST"
HW Version : 0.6
Last Event : HW_EVENT_FAILURE
Last Event Reason : "pon exit <-- UP_WARM_RESET cnt=123 fail_code=LC_POWER_MAIN_FAULT"
sysadmin-vm:0_RP0#
```

过滤受影响的卡的Syslog为了检查日志消息：

```
0/RP0/ADMIN0:Jun 21 00:33:13.487 : cm[1795]: %ROUTING-TOPO-5-OIR_ACTION : OIR card failed
having serial number: SAD173501R7.
0/RP0/ADMIN0:Jun 21 00:33:13.528 : shelf_mgr[1818]: %INFRA-SHELF_MGR-5-CARD_INSERTION :
Location: 0/1, Serial #: SAD173501R7
0/RP0/ADMIN0:Jun 21 00:33:13.528 : shelf_mgr[1818]: %INFRA-SHELF_MGR-6-HW_EVENT : Rcvd HW event
HW_EVENT_FAILURE, event_reason_str 'Initial discovery FAIL: EXIT0, power request on , but not
finish ccc-pon startup. power_control 0x00000001' for card 0/1
0/RP0/ADMIN0:Jun 21 00:33:13.530 : shelf_mgr[1818]: %INFRA-SHELF_MGR-3-CARD_HW_FAILED : Card:
0/1 hardware state going to FAILED
0/RP0/ADMIN0:Jun 21 00:34:06.734 : shelf_mgr[1818]: %INFRA-SHELF_MGR-6-HW_EVENT : Rcvd HW event
HW_EVENT_RESET, event_reason_str 'pon enter --> DOWN_WARM_RESET cnt=3! ' for card 0/1
0/RP0/ADMIN0:Jun 21 00:34:15.987 : shelf_mgr[1818]: %INFRA-SHELF_MGR-6-HW_EVENT : Rcvd HW event
HW_EVENT_POWERED_OFF, event_reason_str 'CCC Warm Reset #8' for card 0/1
0/RP0/ADMIN0:Jun 21 00:34:21.419 : cm[1795]: %ROUTING-TOPO-5-OIR_ACTION : OIR card failed having
serial number: SAD173501R7.
0/RP0/ADMIN0:Jun 21 00:34:21.459 : shelf_mgr[1818]: %INFRA-SHELF_MGR-3-CARD_HW_FAILED : Card:
0/1 hardware state going to FAILED
0/RP0/ADMIN0:Jun 21 00:34:21.459 : shelf_mgr[1818]: %INFRA-SHELF_MGR-6-HW_EVENT : Rcvd HW event
HW_EVENT_FAILURE, event_reason_str 'pon exit <-- UP_WARM_RESET cnt=4
fail_code=LC_POWER_MAIN_FAULT' for card 0/1
```

如果有AssrtHR和DeAssrtHR消息之间的一巨大的时差执行显示history命令的重新启动并且验证。
这很可能是，因为有在VM之间的内部连通性问题。

```
sysadmin-vm:0_RP0#show reboot-history card location 0/1
```

Reset history example: every 20 mins for 2 hours before it recovered:

```
33 0 WarmRst DeAssrtHR 0x00000F32 0x53A4D367 Sat Jun 21 00:35:51 2014
34 0 WarmRst AssrtHR 0x00000F10 0x53A4D81D Sat Jun 21 00:55:57 2014

35 0 WarmRst DeAssrtHR 0x00000F32 0x53A4D821 Sat Jun 21 00:56:01 2014
36 0 WarmRst AssrtHR 0x00000F10 0x53A4DCD7 Sat Jun 21 01:16:07 2014
```

卡德在UNKNOWN状态停留

当卡德报告管理员状态作为未知时，很可能CCC不可能读从板的IDPROM，因此，卡不能完成引导程序。对于这样实例，请执行这些on命令给的位置：

```
RP/0/RP0/CPU0:A41-PE1#show platform
```

Node name	Node type	Node state	Admin state	Config state
0/RP1	NC6-RP	OPERATIONAL	UNKNOWN	
0/FC1	NC6-FC	OPERATIONAL	UNKNOWN	

```
sysadmin-vm:F0_SC0# show controller ccc event-history brief location 0/0
```

```
CCC Card Event History for: 0/0
```

```
Card Event History as seen by Master (0/RP1)
```

```
Current State: CARD_READY
```

DATE	TIME (UTC)	STATE	EVENT
10/02	15:16:54.194	READ_IDPROM	ev_idprom_available
10/02	15:16:53.942	GET_CCC_INFO	ev_get_ccc_info_done
10/02	15:16:53.723	WAIT_ETH_READY	ev_eth_available
10/02	15:16:52.560	CHECK_UBLAZE_BOOT	ev_ublaze_boot_ok
10/02	15:16:52.539	WAIT_CCC_READY	ev_ccc_ready
10/02	15:16:52.537	IDLE	ev_presence_scan

如果CCC在LC运行在引导过程时与使用rconsole，下一步是验证：

1. 在SysadminVM收集进程ID：

```
sysadmin-vm:F0_SC0# show processes ccc_driver location 0/0
```

```
-----  
PID: 2525  
Executable path: /opt/cisco/calvados/packages/ncs6k-sysadmin-boot-5.2.4.CSCut24295  
.all-1.0.0/sbin/ccc_driver  
Instance #: 0  
Respawn: ON  
Respawn count: 1  
Max. spawns per 4 mins: 4  
Last started: 10/02/2015 15:17:23.000  
Process state: Run  
startup_path: /opt/cisco/calvados/packages/ncs6k-sysadmin-boot-5.2.4.CSCut24295  
.all-1.0.0/etc/startup/ccc_driver.startup  
Ready: 5s
```

2. 对LC的Rconsole与使用这些命令：

```
sysadmin-vm:F0_SC0# attach location 0/RP0
```

```
[sysadmin-vm:0_RP0:~]$ exec chvrf 2 bash
```

```
[sysadmin-vm:0_RP0:~]$ chvrf 0 bash
[sysadmin-vm:0_RP0:~]$ /opt/cisco/calvados/sbin/rconsole -l 0/0
Connecting to location 0/0 (backplane-slotid 16, console 0)
Escape sequence is "end"
Waiting for card info from CCC-driver for slot 16
Got card info from CCC-driver for slot 16
IOS Build Date : 04/22/2015 by lchinnad
System Memory Speed : 1334 MHz
Processor Type : Intel(R) Xeon(R) CPU E5-2418L @ 2.00GHz

Press F12 to goto Boot Manager..

Booting System Host OS..
Waiting For CCC Valid Time of Day..
Waiting For CCC Valid Time of Day..
CCC Time: Fri Oct 2 15:16:54 2015

GNU GRUB version 2.00
Press F2 to goto grub Menu..
Booting from Disk..
Loading Kernel..
Loading initrd..
[ 1.949229] i8042: No controller found
Starting udev: [ OK ]
Switching to new root and running init.
Starting udev: [ OK ]
Actual changes:
large-receive-offload: off [requested on]
ntuple-filters: on
Setting hostname host: [ OK ]
Checking filesystems:[ OK ]
Entering non-interactive startup
Bringing up loopback interface: [ OK ]
Bringing up interface eth0: Device eth0 does not seem to be present, delaying initialization.
[FAILED]
Starting system logger: [ OK ]
Starting kernel logger: [ OK ]
Starting kdump:[ OK ]
Starting system message bus: [ OK ]
Starting smartd: [ OK ]
Generating SSH1 RSA host key: [ OK ]
Generating SSH2 RSA host key: [ OK ]
Generating SSH2 DSA host key: [ OK ]
Starting sshd: [ OK ]
Starting xinetd: [ OK ]
Starting crond: [ OK ]
Starting libvirtd daemon: [ OK ]
Starting NCS6k programs for LC on hostos: [ OK ]
mcelog start/running, process 2637
Creating default host password file
serial (/dev/ttyserial (/dev/ttyS1) start/running, process 2649

host login: root
Password:
[host:~]$
[host:~]$
[host:~]$ telnet 0 50001 <-- to get to Calvados
Trying 0.0.0.0...
Connected to 0.
Escape character is '^'.
```

```
sysadmin-vm:0_0 login:
sysadmin-vm:0_0 login: root
Password:
```

```
[sysadmin-vm:0_0:~]$ pgrep ccc <- use pgrep to check if the process is running  
2525
```

```
[sysadmin-vm:0_0:~]$ exit
```

logout

请使用RCONSOLE工具

当卡无法启动时，NCS6008提供给功能执行在卡的远程控制台和发现原因的一个内置的功能为什么卡无法卡住和启动。此功能呼叫RCONSOLE和这是其使用情况示例。

对Rconsole的步骤在特定LC：

1. 导航对SysadminVM
2. 附加对激活RP。
3. 变成全局VRF `chvrf 0 bash`
4. 执行 `/opt/cisco/calvados/sbin/rconsole -l (线路卡)`

示例：

```
RP/1/RP1/CPU0:6008-B#admin  
sysadmin-vm:F0_SC0#  
sysadmin-vm:F0_SC0# attach location 0/RP0 <-- You must be connected to the RP's to be able to  
rconsole  
Tue Oct 20 18:23:54.740 UTC  
[sysadmin-vm:0_RP0:~]$ exec chvrf 2 bash  
[sysadmin-vm:0_RP0:~]$ chvrf 0 bash  
[sysadmin-vm:0_RP0:~]$ /opt/cisco/calvados/sbin/rconsole -l 0/0 & This is LC 0/0  
Connecting to location 0/0 (backplane-slotid 16, console 0)  
Escape sequence is "end"  
Waiting for card info from CCC-driver for slot 16
```

此步骤哪里由TAC是用途广泛确定卡的当前状态和检查其卡住。

命令列表收集，在您开TAC案例前

XR VM：

显示平台

show install激活

Show version

显示重新启动历史记录位置0/0/cpu0

dir misc/disk1

show cli history详细信息

Show log

show tech-support npu

Sysadmin VM：

显示平台详细信息

显示平台片式

show install激活

显示sdr默认sdr重新启动历史记录

显示重新启动历史记录卡位置<>

show controller ccc重置历史记录onbo位置<>

show controller ccc重置历史记录onch位置<>

show controller ccc事件历史记录详细信息位置<>

show tech-support ccc

show tech-support Hbloss

show tech-support slice_manager

show tech-support ctrace

show tech-support sdr_mgr

show log