Contents

Introduction

This document describes troubleshooting techniques for the Nexus 7000 (N7K) hardware.

Debugging Chassis Issues

Fan Issues

This command displays the fan module status on the switch.

Fan status can be one of ok, failure or absent.

- Ok – All fans including the fan controller are functioning properly
- Failure – One or more fans or fan controller have failed. Software cannot determine if a single fan, multiple fans, or all fans have failed. If at least one fan has failed, this status is displayed. This priority 1 syslog message is printed: Fan module Failed.
- Absent – Fan module has been removed. As soon as the fan module is removed, software starts a 5 minute countdown; if the fan module is not re-inserted within 5 minutes, the entire switch is shutdown. Software reads a byte on the Serial Electrically Erasable Programmable Read Only Memory (SEEPROM) to determine if the fan module is present. If the fan module is partially inserted or software is unable to access the SEEPROM on the fan module due to any other reason, software cannot distinguish this case from a real fan module removal. The switch will be shutdown in 5 minutes. If software detects a removal, this priority 0 syslog message is printed every 5 seconds.

- No explicit action is taken by software on a Power Supply fan failure, other than indicating such a failure using syslog messages.

Power Supply

This command displays the power supplies installed, power usage summary and status of power supplies on the switch.

The command as well as a sample output is provided.

Power supply status can be one of these:

- Ok – Power supply is functioning properly
- Fail/Shutdown – Either the power supply has failed or it is shutdown using the switch on the power supply. Whenever a power supply fails, software prints this priority 2 syslog message; Power supply 1 failed or shutdown (Serial number xxxx).
- Shutdown – Software has shutdown the power supply. Software shuts down the lower capacity power supply only if it detects a mis-matched pair of power supplies and the mode is redundant or there is a transition from combined to redundant mode. If both power supplies are the same capacity or the mode is combined, software never shuts down a power supply. This priority 2 syslog message is printed and accompanies a software power supply shutdown; Detected power supply 1. This reduces the redundant power available to the system and can cause service disruptions (Serial number xxxx).
- Absent – The power supply is absent and has been removed. This priority 2 syslog message is printed during a power supply removal; Power supply 2 removed (Serial number xxxx).

Power supply failures:

Each power supply has a LED that indicates power output status. This LED is directly controlled by the power supply and a red color indicates a power supply failure. When you scan the syslog, you might show alternating messages about power supply failure and recovery, further indicating power supply related problems.

Temperature or Heat

Each card in the chassis has atleast two temperature sensors. Each temperature sensor is configured with a minor and a major threshold. This command with sample output shows how temperature information can be retrieved from the switch:

The Intake sensor is placed at the airflow intake and is the most critical indicator of card temperature. All software actions are taken based on a major temperature violation of the Intake sensor.

- All minor threshold violations and major threshold violations on non-Intake sensors

These result in a syslog message, callhome event and a Simple Network Management Protocol (SNMP) trap. This priority 1 or 2 messages are printed in the syslog – Module 1 reported Major temperature alarm (sensor-index 1 temperature 76).

- Major temperature threshold violation on a linecard on Intake sensor

The linecard is instantly shutdown with this priority 0 syslog message - Module 1 powered down due to major temperature alarm.

- Major temperature threshold violation on a redundant Supervisor on Intake sensor

The redundant Supervisor is instantly shutdown. This will result in either a switchover or the standby shutting down, depending on the particular Supervisor that violated the threshold. This priority 0 syslog message is displayed - Module 1 powered down due to major temperature alarm.

- Temperature sensor failure

Sometimes, the temperature sensors fail and become inaccessible. No explicit software action is taken for this condition. This priority 4 syslog message is printed – Module 1 temperature sensor failed.

Debugging Supervisor Module Issues

Switch/Supervisor Reset/Reload

Debugging a switch/supervisor level reset/reload typically involves looking into debug/log information stored on the Non-Volatile Random Access Memory (NVRAM) on the Supervisors. There are 3 kinds of debug/log information present in the NVRAM that might hold some important information.

1.1 Reset reason

Reset reasons are stored on the Supervisor NVRAM on each Supervisor. Each Supervisor stores its own reset reason. After the switch comes back up, the reset reasons can be dumped using this CLI command. A sample output is provided.

Upto the last 4 reset reasons are saved and displayed. A reset reason contains:

- Timestamp of when the reset/reload occurred
- Reason for resetting/reloading the card
- Service that caused hat reset/reload – if any
- Software version that was running at that time

Sometimes a reset reason of Unknown is displayed. Reset reasons that are unknown to software or beyond software control are categorized as Unknown. These typically include:

- Any power cycle of switch – including controlled power cycle of power supplies or a reset of power supplies caused by a power glitch or power failure
- Front panel reset button reset on Supervisor
- Any other hardware failures causing the CPU/DRAM/IO to reset or hang

1.2 NVRAM syslog

Syslog messages that are priority 0, 1 and 2 are also logged into the NVRAM of the Supervisor. After the switch comes back up online, syslog messages in the NVRAM can be displayed using this command. The command and a sample output is displayed:

Scanning the NVRAM syslog might provide some more information on the particular failure that caused the switch/Supervisor reload/reset.

1.3 Module exceptionlog

Module exceptionlog is a wraparound log of all errors and exceptional conditions on each module. Some exceptions are catastrophic, some partially affect certain ports in a module, others are for warning purposes. Each log entry has the particular device that logged the exception, the exception level, error code, ports affected, timestamp. The exception log is stored in the NVRAM on the Supervisor and it can be displayed using this CLI command. A sample output is provided.

The exceptionlog provides critical information to troubleshoot errors and exception conditions. Some of the device IDs are listed here.

In the Multilayer Data Switch (MDS) Chassis, the supervisor modules are brought up a little differently than the line-card modules. When two supervisors are present in the system and the system is powered-up, one of the supervisors will become active and the other standby. Active Supervisor bring-up and Standby Supervisor bring-up is different and is discussed here.

Active Supervisor Bring-up

If there is no active supervisor in the system, the supervisor which boots up will default to active supervisor. A process called system manager is responsible for loading all the software components in an orderly fashion on the supervisor. One of the first software components that is run on the supervisor is the platform manager. This component will load all the kernel drivers and handshakes with the system manager. On Success, system-manager will go ahead and start the rest of the processes based on the internal dependency between processes.

From module manager’s perspective, Supervisor is just like another line-card module with subtle differences. When platform manager indicates to module manager that the Supervisor is UP, module manager does not wait for Registration. Instead, it informs all the software components that Supervisor is up (also known as Sup Insertion Sequence). All the components will configure the supervisor. If any component comes back with a failure, the supervisor will be rebooted.

Standby Supervisor Bring-up

If there is an active supervisor in the system, the supervisor which is booting up will default to standby supervisor state. The standby supervisor needs to mirror the state of the active supervisor. This is achieved by ‘system manager’ on active, initiating a gsync (global sync) of active supervisor state to standby supervisor. Once all the components on the standby are synchronized with that of the active supervisor, module manager is informed that the standby supervisor is up.
Module-manager will now go ahead and inform all the software components on the active supervisor to configure the standby supervisor (Also Known as Standby Sup Insertion Sequence). Any errors from any component during the Standby Sup Insertion Sequence will result in Standby Supervisor Reboot.

Active Supervisor Reboot

MDS maintains lot of debug information during runtime. But, whenever a supervisor reboots much of the debug information is lost. However all critical information is stored in non volatile ram, which can be used to reconstruct the failure. When an Active Supervisor reboots, the information that is stored in its nvram cannot be obtained until it comes back up again. Once the Supervisor comes back up again, these commands can be used to dump the persistent log:

Switch# show logging nvram
Switch# show system reset-reason
Switch# show module internal exception-log

Example 1: Active Sup Reboot (due to Supervisor Process Crash)

In this example, a Supervisor Process crashed (Service “xbar”) which causes the Active sup to be rebooted. When the supervisor comes back up again, the information stored in the reset-reason gives a clear indication, for the reboot of the supervisor.

If there is standby supervisor in the system, the standby supervisor will now become active supervisor. Displaying the syslog information on the standby supervisor will also provide the same information (although not as explicitly as ‘show system reset-reason’).

Example 2: Active Sup Reboot (due to runtime diagnostic failure)

In this example, Supervisor in slot-6 is active and the arbiter on the Supervisor reports a Fatal Error. When any hardware device reports a Fatal Error, the module that contains the device is rebooted. In this case the Active Supervisor is rebooted. If there is a standby supervisor, the standby supervisor will take over. Syslog messages on the standby supervisor and exception log will have information to identify the source of error.

In addition, when the rebooted sup comes online again, ‘**show system reset-reason**’ will contain relevant information too. In this case the module 6 (which was the active sup) was rebooted by Sap 48 with error-code 0x80000020. The process which owns this sap can be obtained by the command ‘show system internal mts sup sap 48 description’ which says that the process was xbar-manager.

Example 3: Standby Sup Failed to Come Online

In this example, active sup is up and running and standby sup is plugged into the system. However **show module** does not indicate that the module has ever come up.

However, if you login to the console of the standby sup, it says it is standby.

As discussed earlier, when the standby sup is inserted into the system, the configuration and the state of all the components of the active supervisor is copied over to the standby (gsync). Till this process is complete, active supervisor does not consider standby supervisor is present. To verify if this process is complete, you could issue the following command on the active supervisor. The output of the command indicates that synchronization in progress (and is probably never completed).

The most likely reason why this could have happened is, if one of the software components on the standby failed to synchronize its state with the active supervisor. To verify which processes did not synchronize, you can issue this command on the active supervisor and the output indicates a lot of software components have not completed gsync.

In addition, looking at the standby supervisor we see that xbar software component has been restarted 23 times. This looks like the most likely cause that the standby did not come up.

Example 3: Standby Sup is in Powered-up State

In this example, standby sup is inserted in slot 6. **show module** command issued on the active-sup, shows Standby Sup is in powered-up state.

In this example, **show logging** does not give any valuable information and neither does **show module internal exception-log**. However as all state transitions for a given module is stored in the module manager we can look at the state transistions of the module manager to figure out what is wrong. The internal state transistions are:

Look at the logs above Index 92, indicates that the supervisor is in failed state and the triggered event is LCM_EV_LC_INSERTED_SEQ_FAILED. (Insertion sequence failed). Going up the logs to find out why Insertion Sequence failed, see that insertion sequence failed right after a response from MTS_SAP_XBAR_MANAGER (Index 73 and Index 74). This indicates that there is something wrong with xbar configuration when the standby sup is inserted. More debugging can be done by looking at the internal logs of the failed component (in this case, xbar component).