

Устранение проблем 3650/3850 перезагружается менеджером стека через системный отчёт

Введение

Устранение проблем повторных загрузок стека через системный отчёт в отсутствие катастрофического отказа обычно делается на платформах коммутации NGWC с помощью stackwise технологию. Текущая документация ограничена на использовании системного отчёта, и это руководство пишется, чтобы объяснить, как можно усилить эти отчёты диагностировать проблемы, как правило, найденные с укладкой проблем. Это руководство особенно приспособлено для Catalyst 3650/3850 архитектуры коммутации рабочий XE IOS, который поддерживает возможности укладки.

Большинство проблем с stackwise технологией происходит от проблемы связи между участниками в стеке. Любое несоответствие информации между участниками или потерей подключения может привести к проблеме, которая проникает через весь стек, в конечном счете приводящий к сбросу с менеджером стека. Этот документ выделит некоторые наиболее распространенные типы сбоев, замеченных с повторными загрузками менеджера стека, использованием системного отчёта и соответствующими CLI, доступными для диагностирования и различные типы проблем.

Общие сведения:

Системный отчёт по сравнению с отчетами коммутаторов

Системный отчёт является всесторонним отчетом об участнике от того, как он чувствует состояние стека. Это не crashinfo (сведения об аварийном отказе) (который разгрузит память для дальнейшей отладки), но вместо этого, отчёт, который имеет журналы и информацию об отладке для различных сервисов, работающих под XE IOS, который был бы полезен для разработки для отслеживания состояния того сервиса. Системный отчёт может генерироваться, когда коммутатор повторно загружен менеджером стека, катастрофический отказ процесса произошел, или вручную генерировал пользователем во время оперативной операции.

Во многих случаях существуют ситуации, в которых один коммутатор мог бы отказать в стеке, но остаток участников может остаться неповрежденным. Для сбора как информация о состоянии стека в данном времени switch_reports были представлены так, чтобы остающиеся участники генерировали тот, когда это обнаружит, что выключился участник. switch_report будет локальным отчётом того, как тот участник чувствует текущее состояние стека.

Примечание: Эти отчеты написаны и сжаты так, они не могут быть распечатаны к терминалу с помощью 'больше'. Они должны будут быть переданы от коммутатора и распакованы для просмотра журнала.

Где собрать систему/отчеты коммутаторов

Системные отчеты будут, как правило, писаться в crashinfo (сведения об аварийном отказе): каталог участника в том стеке. Например, если будет стек х-коммутатора-участника, то каждый коммутатор будет иметь их собственный каталог crashinfo (сведения об аварийном отказе), который может быть доступным использованием "crashinfo-(сведения-об-аварийном-отказе)-x dir", где 'x' соответствует тому участнику в стеке.

```
3850#dir crashinfo 1:
```

```
Каталог crashinfo:/
```

```
11 -          rwx 355 14 августа 2015 7:48:17 -  
04:00 last_systemreport_log
```

```
12 -          rwx 724015 15 октября 2014 7:14:32 - 04:00 system-  
report_1_20141015-111342-UTC.gz
```

```
3850#dir crashinfo 2:
```

```
Каталог crashinfo-2:/
```

```
11 -          rwx 357 14 августа 2015 7:50:49 -  
04:00 last_systemreport_log
```

```
12 -          rwx 751340 15 октября 2014 6:41:12 - 04:00 system-  
report_1_20141015-104022-UTC.gz
```

Примечание: Обязательно соберите выходные данные для "crashinfo-(сведения-об-аварийном-отказе)-x dir": для каждого коммутатора в том стеке, потому что 'покажите технологию' не перечислит доступные файловые системы или файлы crashinfo. Важно, чтобы у вас было все изображение каждого участника в том стеке. Обновление: С более новых версий XE IOS> 3.6E, технология показа отразит "crashinfo (сведения об аварийном отказе) dir": + 'выходные данные show file systems. См. [CSCun50428](#) .

Содержательные разделы в системном отчёте

С точки зрения TAC это некоторые более обычно просматриваемые записи в рамках системного отчёта, который может помочь диагностировать события конкретного вопроса. Существуют другие журналы от других сервисов, содержащихся в здесь, что разработка может найти, хотя и рассмотреть.

файл журнала: /mnt/pss/sup_sysmgr_reset.log

Это - короткие выходные данные к очень, в общем понимают, почему был замечен сброс. Посмотрите ниже типов раздела сбоев для взгляда значения и контекста в том, как будут варьироваться эти причины.

файл журнала: IOS

Это - буфер журнала, поддержанный из IOSd. Любые команды, которые были выполнены пользователем или системными журналами, генерируемыми в IOSd, будут найдены в этом разделе. Новые журналы находятся к концу этих выходных данных.

Буфер трассировки: stack-mgr-events

Отслеживает события, замеченные от менеджера стека, который будет включать, когда другие участники присоединятся/удалят от стека или какой порт стеков сообщения входят через.

Буфер трассировки: redundancy-timer-ha_mgr

События поддержки активности показов между коммутаторами в стеке. Когда отказ в связи запустился, метки времени могут помочь определять.

Типы сбоев

Этот раздел выделит некоторый обычно замечаемый сброс из системного отчёта, который, как правило, вызывается процессом менеджера стека. Менеджер стека является процессом Linux, который управляет участниками в стеке и будет наблюдать за любыми изменениями в ролях между коммутаторами в стеке. Если менеджер стека обнаружит проблему во время инициализации или выборов роли, то она передаст сигнал повторной загрузки к отдельным коммутаторам для стека для сброса. Ниже также перечислит известные ошибки, которые были привязаны к соответствующему типу отказа.

Примечание: Не все повторные загрузки менеджера стека приписаны неполадкам программного обеспечения. Фактически, более распространено рассмотреть эти проблемы декларация как вторичную проблему / проблему жертвы к проблеме используемого оборудования.

Сброс Reason:Reset/Reload запрашивает [менеджер стека]. [Несовместимость ISSU]

Когда существует объемный синхронизирующий сбой при попытке синхронизировать конфигурацию на ведущем устройстве между всеми участниками в стеке, вы могли бы видеть этот тип сброса. Проверка файла журнала: IOS или журналы от активного коммутатора могли бы выделить конфигурации, которые способствовали этому сбросу.

Перезагрузите изодромный с предварением Reason:Service [iosd]: [xyz] завершился неправильно [11].

Замеченный, когда коммутатор завершается катастрофическим отказом в процессе IOSd. При рассмотрении каталогов crashinfo (сведения об аварийном отказе) для любых файлов crashinfo + дампы основной памяти могут использоваться для отладки этого сбоя далее.

hap_sup_reset: Сброс Reason:Reset/Reload запрашивает [менеджер стека]. [сложите слияние]

Слияние стека замечено, когда существует два или больше коммутатора, которые полагают, что они - мастер стека. Это может быть замечено, когда существует перерыв в вызове стека (т.е. два кабеля разъединены от стека), таким образом, и активное и резервное теряют связь другим участникам. Добавление уже приведенного в действие коммутатора к существующему стеку может вызвать слияние стека, поскольку будет два активных коммутатора в стеке.

[CSCuh58098](#) - Когда проблемы разводки кабелей стека присутствуют, 3850 стеков могут перезагрузиться

[CSCui56058](#) - Включение логики устранения дребезжания для кабеля стека

[CSCup53338](#) - 3850 IOSD завершаются катастрофическим отказом | Signal=SIGSEGV (11) pm_port_data_from_swidb

hap_sup_reset: Код причины: [4] Сброс Reason:Reset/Reloadrequested [менеджером стека]. [стек объединяется из-за несовместимости]

Когда активное и резервный коммутатор существуют в стеке, это было замечено в ситуациях. Если активный коммутатор теряет связь резерву, то резерв попытается вступить во владение как активное даже при том, что активное подключено все еще.

[CSCuo49555](#) /CSCup58016 - 3850/3650 завершается катастрофическим отказом должный одноадресно передать лавинную рассылку на порту mgmt

[CSCur07909](#) - Слияние стека из-за активного и резервного потерянного подключения

Сброс Reason:Reset/Reload запрашивает [менеджер стека]. [Неправильный соседний узел встретился после избирательного бюллетеня ASIC]

Коммутаторы участвуют в избирательном бюллетене ASIC во время, загружаются для определения его соседних коммутаторов в стеке. Этот сброс может быть замечен, когда таймер истекает для соседнего узла для объявления себя или если существует ошибка логики во время nbrcast диалога. Это было замечено в контексте неисправных кабелей стека, которые были решены через замену.

[CSCun60777](#) - Коммутатор перезагрузился из-за Неправильного соседнего узла, с которым встречаются после избирательного бюллетеня ASIC

[CSCud93761](#) - Коммутатор перезагрузился из-за Неправильного соседнего узла, с которым встречаются после избирательного бюллетеня ASIC

hap_sup_reset: Код причины: [4] Сброс Reason:Reset/Reload запрашивает [менеджер стека]. [потерянный и активный и резервный]

Это, как правило, замечается от участника на стеке, который не находится в или активное или роль резерва (standby). Когда активные сбои, если не будет никакого резервного коммутатора для принятия активной роли для стека, то весь стек перезагрузит. Если стек является нестабильным состоянием, или политика увольнение еще не синхронизировали, это может быть замечено. Это вероятно жертва, почему активное / резервные коммутаторы выключились или потенциально индикация, которую резервирование не синхронизирует правильно. Это может также быть замечено в том, когда стеки настроены в полукольцевой настройке.

[CSCup53882](#) - Коммутаторы - участники в 3850 перезагрузках стека - [потерянный и активный и резервный]

hap_sup_reset: Код причины: [1] Сброс Reason:Reset/Reload запрашивает [менеджер стека]. [Keepalive_Timeout]

Замеченный, когда сообщения поддержки активности не получены от коммутатора в стеке. Рассмотрение “Буфера трассировки: redundancy-timer-ha_mgr” должен показать обмен сообщениями поддержки активности и предоставить перспективу времени для того, когда начался отказ в связи. Собираение отчетов коммутаторов от остатка стека и рассмотрение журналов в течение выделенного интервала времени могут помочь здесь.

hap_sup_reset: Сброс Reason:Reset/Reload запрашивает [менеджер стека]. [Команда Reload]

Это - довольно интуитивная причина сброса – это замечено, когда менеджер стека получает запрос повторной загрузки, который мог быть вызван через CLI или внешне через устройство управления (SNMP). В случаях [CSCuj17317](#) , это также обнаружится как ‘команда повторной загрузки’, выполненная также. От файла журнала: IOS вы видите:

```
CMD: 'reload'  
%SYS-5-RELOAD: Reload requested by console. Reload Reason: Reload command.  
%STACKMGR-1-RELOAD_REQUEST: 1 stack-mgr: Received reload request for all switches, reason Reload  
command  
%STACKMGR-1-RELOAD: 1 stack-mgr: Reloading due to reason Reload command
```

Соответствующие дефекты

[CSCur76872](#) - Когда система исчерпывает буфер SDP, менеджер стека выключается.

[CSCup49704](#) - 3850 Катастрофических отказов FED - Ждуций каналов SPI FED_SPI_FLCD, FED_SPI_FAST...

Диагностирование потенциального кабельного подключения/проблемы порта стека

Признак 1) Любые знаки проблемы разводки кабелей стека будет очевиден любой переброской порта стеков до сброса. Рассмотрение “файла журнала: IOS” отчёт до сброса, как правило, является отличным местом для начала. Вот пример того, где вы видите переброску порта стеков, который зарегистрирован и на текущем SW2 и на резервном SW1. Этот тот же порт стеков махал каждым в каждом экземпляре сброса и был решен путем замены кабеля стека:

```
===== log file: IOS =====  
.  
.  
Aug 8 21:40:14.532 UTC: %STACKMGR-1-STACK_LINK_CHANGE: STANDBY:1 stack-mgr: Stack port 1 on  
switch 1 is down (SW1-1)  
Aug 8 21:40:17.242 UTC: %STACKMGR-1-STACK_LINK_CHANGE: STANDBY:1 stack-mgr: Stack port 1 on  
switch 1 is up (SW1-1)  
Aug 8 21:46:11.194 UTC: %STACKMGR-1-STACK_LINK_CHANGE: 2 stack-mgr: Stack port 2 on switch 2 is  
down  
Aug 8 21:46:12.937 UTC: %STACKMGR-1-STACK_LINK_CHANGE: 2 stack-mgr: Stack port 2 on switch 2 is  
up  
Aug 8 21:48:23.063 UTC: %STACKMGR-1-STACK_LINK_CHANGE: 2 stack-mgr: Stack port 2 on switch 2 is  
down  
Aug 8 21:48:24.558 UTC: %STACKMGR-1-STACK_LINK_CHANGE: 2 stack-mgr: Stack port 2 on switch 2 is  
up  
Aug 8 21:50:40.666 UTC: %STACKMGR-6-SWITCH_REMOVED: 2 stack-mgr: Switch 1 has been removed from  
the stack.  
Aug 8 21:50:40.671 UTC: Starting SWITCH-DELETE sequence, switch 1
```

Признак 2) В зависимости от настройки stackwise используется (180, 480, плюс), количество вызовов передачи на ASIC порта будет варьироваться. Эти команды опросят глобальные регистры, которые поддерживают промежуточную сумму того, сколько ошибок чтения замечено для каждого вызова передачи. ‘Asic порта 0’ соответствует порту стеков 1, и ‘asic порта 1’ соответствует порту стеков 2. Это должно быть выполнено для каждого коммутатора, и любые знаки приращения количества могут изолировать ли там, возможно, проблема в порту или с кабелем стека.

Они могут собираться несколько раз за несколько минут для сравнения дельт в количестве:

show platform port-asic <0-1> регистр чтения коммутатор SifRacDataCrcErrorCnt <switch#>

- Сегмент с ошибкой CRC данных

show platform port-asic <0-1> регистр чтения коммутатор SifRacRwCrcErrorCnt <switch#>

- Инкрементно увеличенный на любой отказавшей проверке CRC

show platform port-asic <0-1> регистр чтения коммутатор SifRacPcsCodeWordErrorCnt <switch#>

- Инкрементно увеличенный на недопустимом коде PCS, неизвестном кодовом слове PCS, рабочая ошибка несоизмеримости обнаружена

show platform port-asic <0-1> регистр чтения коммутатор SifRacInvalidRingWordCnt <switch#>

- Ошибка в канале связи на стеке вызвала ringword ошибку CRC

Ниже приводится пример, где у вас было событие слияния стека, замеченное оба участника стека с 2 участниками без любых знаков колеблющегося порта стеков. Вы видите, что вызов [0] инкрементно увеличивает с CRC на порте стеков 1 из коммутатора 1, и закончили тем, что заменили кабель стека для заканчивания этой проблемы.

```
3850#$show platform port-asic 0 read register SifRacRwCrcErrorCnt switch 1
Load for five secs: 11%/4%; one minute: 11%; five minutes: 12%
Time source is NTP, 14:02:49.119 EDT Thu Aug 20 2015
```

For asic 0

```
SifRacRwCrcErrorCnt on Asic 0
[0]
      count 0x00000031 <<
[1]
      count 0x00000001
[2]
      count 0x00000000
[3]
      count 0x00000001
[4]
      count 0x00000000
[5]
      count 0x00000001
```

```
3850#$show platform port-asic 0 read register SifRacRwCrcErrorCnt switch 1
Load for five secs: 9%/4%; one minute: 11%; five minutes: 12%
Time source is NTP, 14:02:53.550 EDT Thu Aug 20 2015
```

For asic 0

```
SifRacRwCrcErrorCnt on Asic 0
[0]
      count 0x000000c9 <<
[1]
      count 0x00000001
[2]
      count 0x00000000
[3]
      count 0x00000001
[4]
      count 0x00000000
[5]
      count 0x00000001
```


Примечание: В зависимости от регистра, на который смотрят, маска, возможно, другая в каждом случае. В вышеупомянутом примере маска повторится на последних 14 битах. Таким образом, когда счетчик достигает 0x00003FFF, он перенесется назад к 0x00000000.

Дополнительные советы

1. Архивация каталогов crashinfo (сведения об аварийном отказе)

Больше коммутаторов в стеке означает, что будет больше файлов отчета, которые будут собраны. Легко быть разбитым количеством отчетов, которые генерируются. Организация жизненно важна для изоляции сбоя. Найдите непротиворечивость с помощью меток времени того, когда каждый коммутатор записал файл отчета для данного инцидента, если это возможно. Оттуда, попросите те очень определенные отчеты из тех коммутаторов данных так, клиент не загружает несколько файлов. Каталог crashinfo (сведения об аварийном отказе) может также быть заархивирован так, клиент может передать одиночный архив, содержащий заинтересованные отчеты. Придерживающееся создаст архив, названный 'crashinfo-(сведения-об-аварийном-отказе)-archive.tar' во флэше - каталоге:

```
F340.03.10-3800-1#archive tar /create ?  
WORD Tar filename
```

```
F340.03.10-3800-1#archive tar /create crashinfo-archive.tar ?  
WORD Dir to archive files from
```

```
F340.03.10-3800-1#archive tar /create crashinfo-archive.tar crashinfo ?  
WORD File or Dir  
<cr>
```

```
F340.03.10-3800-1#archive tar /create crashinfo-archive.tar crashinfo:
```

2. Восстановление нестабильного стека

Могут быть некоторые ситуации, где вы видите несколько, участники в стеке, перезагружающемся во время, загружаются после того, как процесс голосования стека имеет место. Если перезагруженный коммутатор полагает, что себя мастер стека тогда, это может часто приводить к событию слияния стека и введет в состояние цикла загрузки. В этой ситуации может быть желательно спросить клиента:

- Выключите весь стек и переустановите все кабели стека твердо.

- Включение питания каждый коммутатор - участник в стеке один за другим, пока все участники не сошлись к его ожидаемому состоянию.

- В случаях, где участник не в состоянии присоединиться к стеку, удалите это из стека и попытайтесь загрузить это частное лицо как автономное для устранения проблем далее.

3. Генерируйте системные отчёты вручную

Вручную создание системных отчётов требует, чтобы был включен 'внутренний сервис'. Это напишет системный отчет как текстовый файл, который может быть сделан на основании коммутатора.

```
3800-1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
3800-1(config)#service internal
3800-1(config)#exit

3800-1#resource create_system_report ?
WORD system report filename

3800-1#resource create_system_report sysreport.txt ?
switch Switch number
<cr>

3800-1#resource create_system_report sysreport.txt switch ?
<1-1> Switch number

3800-1#resource create_system_report sysreport.txt switch 1 ?
<cr>
```