

ASR5x00 세션 관리자 작업 - 기능, 충돌, 복구 작업 및 충돌 로그에 대한 설명

목차

[소개](#)

[소프트웨어 아키텍처:복원력을 위해 설계](#)

[충돌이란 무엇입니까?](#)

[세션 관리자 크래시의 영향](#)

[교환원은 언제 걱정해야 합니까?](#)

[충돌이 발생했는지 확인하는 방법](#)

[충돌 로깅 아키텍처](#)

[관리 카드 간 충돌 이벤트 및 미니코어 동기화](#)

[명령](#)

[요약](#)

소개

이 문서에서는 Cisco ASR(Aggregation Services Router) 5x00 Series의 소프트웨어 신뢰성, 서비스 가용성 및 장애 조치 기능에 대해 설명하고 설명합니다. ASR5x00의 소프트웨어 충돌 및 소프트웨어 충돌의 영향을 나타냅니다. 이 문서는 예기치 않은 소프트웨어 충돌이 발생하더라도 ASR5x00이 내재된 소프트웨어 복원력 및 가용성 기능으로 인해 "캐리어급" 가용성의 목표를 어떻게 제공할 수 있는지 설명합니다. 가입자가 서비스 가용성에 대해 생각할 필요는 없다. Cisco의 목표는 단일 하드웨어 또는 소프트웨어 장애로 인한 세션 손실이 아니라, 즉 전체 시스템의 손실, 즉 음성 등급 신뢰성입니다. ASR5x00의 소프트웨어 안정성 기능은 운영자의 네트워크에서 예기치 못한 장애가 발생할 수 있는 경우에도 "통신업체 등급" 서비스 가용성 목표를 달성할 수 있도록 설계되었습니다.

소프트웨어 아키텍처:복원력을 위해 설계

ASR5x00에는 다양한 특정 기능을 수행하도록 설계된 PSC(Packet Services Card) 또는 DPC(Data Processing Card) 및 SMC(System Management Card) 또는 MIO(Management and I/O) 카드에 분산된 소프트웨어 작업 모음이 있습니다.

예를 들어, 세션 관리자 작업은 가입자 집합에 대한 세션을 처리하고 사용자 트래픽에서 P2P(Peer-to-peer), DPI(Deep Packet Inspection) 등과 같은 인라인 서비스를 수행합니다. AAA(Authentication, Authorization, and Accounting) 관리자 작업은 가입자 트래픽 사용량 등을 기록하기 위해 청구 이벤트를 생성하는 작업을 담당합니다. 세션 관리자 및 AAA 관리자 작업은 PSC/DPC 카드에서 실행됩니다.

SMC/MIO 카드는 O&M(Operation and Maintenance) 및 플랫폼 관련 작업에 예약되어 있습니다. ASR5x00 시스템은 가입자 세션 처리를 위한 세션 하위 시스템 및 IP 주소 할당, 라우팅 등을 담당하는 VPN 하위 시스템과 같은 서로 다른 소프트웨어 하위 시스템으로 가상으로 구분됩니다. 각 하

위 시스템에는 제어 하위 시스템의 상태를 감독하는 컨트롤러 작업이 있습니다. 컨트롤러 작업은 SMC/MIO 카드에서 실행됩니다. 세션 관리자 및 AAA 관리자 작업은 제어, 데이터 트래픽 및 청구 목적으로 가입자의 세션을 처리하기 위해 함께 피어링됩니다. 시스템에서 세션 복구가 활성화되면 각 세션 관리자 작업은 세션 관리자 충돌이 발생할 경우 복구할 피어 AAA 관리자 작업과 함께 가입자 상태 집합의 상태를 백업합니다.

충돌이란 무엇입니까?

정상 작동 중에 오류 조건이 발생할 경우 ASR5x00의 작업이 잠재적으로 충돌할 수 있습니다. ASR5x00의 크래시 또는 소프트웨어 결함은 시스템에서 *예상치 못한* 종료 또는 작업 종료로 정의됩니다. 소프트웨어 코드가 금지된 메모리 영역(예: 손상된 데이터 구조)에 액세스하려고 하거나, 예기치 않은 코드(예: 잘못된 상태 전환)에 문제가 발생하는 경우 충돌이 발생할 수 있습니다. 시스템 모니터 작업에 작업이 응답하지 않고 모니터가 작업을 중단하고 다시 시작하려고 하면 충돌이 트리거될 수도 있습니다. 작업 상태를 분석하기 위해 작업이 CLI 명령 또는 시스템 모니터로 현재 상태를 덤프하도록 강제할 때 시스템에서 충돌 이벤트를 명시적으로 트리거할 수도 있습니다(예상치 못한 것과 반대). 시스템 컨트롤러 작업이 다시 시작되어 반복적으로 실패한 관리자 작업을 통해 상황을 잠재적으로 수정하려는 경우에도 예상 충돌 이벤트가 발생할 수 있습니다.

세션 관리자 크래시의 영향

정상 작동 시 세션 관리자 작업은 해당 가입자 세션에 대한 청구를 처리하는 피어링 AAA 관리자 작업과 함께 세션에 대한 가입자 세션 및 관련 데이터 트래픽 집합을 처리합니다. 세션 관리자 충돌이 발생하면 시스템에 존재하지 않게 됩니다. 시스템에서 세션 복구가 활성화된 경우, 동일한 PSC/DPC 카드에서 대기 세션 관리자 작업이 활성화됩니다. 이 새 세션 관리자 작업은 피어 AAA 관리자 작업과 통신할 때 가입자 세션을 복원합니다. 복구 작업의 범위는 50msec에서 몇 초 사이이며, 이는 충돌 시 세션 관리자에서 활성 상태였던 세션 수와 카드의 전체 CPU 로드 등에 따라 달라집니다. 이 작업의 원래 세션 관리자에서 이미 설정된 가입자 세션의 손실이 없습니다. 사고 당시 가입이 진행 중이었던 가입자 등록도 프로토콜 재전송 등으로 복구될 것으로 보인다. 충돌 시 시스템을 통해 전환된 모든 데이터 패킷은 네트워크 연결의 통신 엔티티에 의한 네트워크 손실과 관련이 있다고 가정할 수 있으며 재전송되고 새 세션 관리자가 연결을 수행합니다. 세션 관리자가 수행한 세션에 대한 청구 정보는 피어 AAA 관리자에서 유지됩니다.

교환원은 언제 걱정해야 합니까?

세션 관리자 충돌이 발생하면 복구 절차는 이전에 설명한 대로 수행되며 나머지 시스템은 이 이벤트의 영향을 받지 않습니다. 한 세션 관리자에서 충돌이 발생해도 다른 세션 관리자에는 영향을 미치지 않습니다. 운영자에 대한 지침으로서, 동일한 PSC/DPC 카드에 여러 세션 관리자 작업이 동시에 또는 10분 이내에 충돌할 경우 시스템이 새 세션 관리자를 빠르게 시작하지 못할 수 있으므로 세션이 손실될 수 있습니다. 이는 세션 손실이 발생할 수 있는 이중 결함 시나리오에 해당합니다. 복구가 가능하지 않을 경우 세션 관리자는 단순히 다시 시작되며 새 세션을 받아들일 준비가 됩니다.

지정된 세션 관리자가 반복적으로 crash(예: 동일한 결함 조건이 반복적으로 발생함)하면 세션 컨트롤러 작업은 메모를 수행하고 하위 시스템을 복원하기 위해 자동으로 다시 시작됩니다. 세션 컨트롤러 작업이 세션 하위 시스템을 안정화할 수 없고 이 작업을 통해 계속 다시 시작할 경우, 에스컬레이션의 다음 단계는 시스템이 대기 SMC/MIO 카드로 전환하는 것입니다. 대기 SMC/MIO 카드가 없거나 전환 작업에서 오류가 발생하면 시스템이 자동으로 재부팅됩니다.

또한 세션 관리자는 각 APN(Access Point Name), 서비스, 기능 등에 대한 통계를 유지 관리하므로 충돌이 발생하면 영구적으로 손실됩니다. 따라서 블록통계를 주기적으로 수집하는 외부 엔티티는 하나 이상의 충돌이 발생할 때 통계에 드롭을 관찰합니다. 이는 시간 축 위에 그려진 통계를 그래픽으로 표시하는 딥(dip)으로 나타낼 수 있습니다.

참고: 7-14 PSC 또는 4-10 DPC 카드가 장착된 일반적인 새시에는 PSC/DPC 카드 수에 따라 약 120-160 명의 세션 관리자가 있으며, 단일 충돌이 발생하면 통계의 약 1/40 또는 1/80^이 손실됩니다. 대기 세션 관리자가 작업을 인계하면 0부터 다시 통계를 누적하기 시작합니다.

충돌이 발생했는지 확인하는 방법

충돌이 발생하면 EMS(이벤트 모니터링 서비스) 및 syslog 이벤트와 같은 네트워크 모니터링 스테이션에 SNMP 트랩 이벤트가 트리거됩니다. 시스템에서 발생한 충돌을 **show crash list** 명령으로 볼 수도 있습니다. 이 명령은 앞서 설명한 대로 예기치 않은 충돌 이벤트와 예상 충돌 이벤트를 모두 나열합니다. 이러한 두 가지 유형의 충돌 이벤트는 각 충돌을 설명하는 헤더를 통해 구별할 수 있습니다.

작업 충돌 후 세션 복구가 성공하면 다음 로그 메시지가 표시됩니다.

```
"Death notification of task <name>/<instance id> on <card#>/<cpu#> sent to parent task <parent name>/<instance id> with failover of <task name>/<instance id> on <card#>/<cpu#>"
```

복구할 수 없는 작업 충돌이 다음 로그 메시지로 표시됩니다.

```
"Death notification of task <name>/<instance id> on <card#>/<cpu#> sent to parent task <parent name>/<instance id>"
```

요약하자면, 세션 복구가 활성화되면 대부분의 경우 가입자에 영향을 주지 않으므로 충돌이 감지되지 않습니다. CLI 명령을 입력하거나 로그 또는 SNMP 알림을 확인하여 충돌 발생을 탐지해야 합니다.

예를 들면 다음과 같습니다.

```
***** show crash list *****
Tuesday May 26 05:54:14 BDT 2015
=== =====
# Time Process Card/CPU/ SW HW_SER_NUM
PID VERSION MIO / Crash Card
=== =====

1 2015-May-07+11:49:25 sessmgr 04/0/09564 17.2.1 SAD171600WS/SAD172200MH
2 2015-May-13+17:40:16 sessmgr 09/1/05832 17.2.1 SAD171600WS/SAD173300G1
3 2015-May-23+09:06:48 sessmgr 03/1/31883 17.2.1 SAD171600WS/SAD1709009P
4 2015-May-25+15:58:59 sessmgr 09/1/16963 17.2.1 SAD171600WS/SAD173300G1
5 2015-May-26+01:15:15 sessmgr 04/0/09296 17.2.1 SAD171600WS/SAD172200MH

***** show snmp trap history verbose *****
Fri May 22 19:43:10 2015 Internal trap notification 1099 (ManagerRestart) facility
sessmgr instance 204 card 9 cpu 1
Fri May 22 19:43:29 2015 Internal trap notification 73 (ManagerFailure) facility
sessmgr instance 204 card 9 cpu 1
Fri May 22 19:43:29 2015 Internal trap notification 150 (TaskFailed) facility
```

sessmgr instance 204 on card 9 cpu 1
Fri May 22 19:43:29 2015 Internal trap notification 151 (TaskRestart) facility
sessmgr instance 204 on card 9 cpu 1
Fri May 22 19:43:30 2015 Internal trap notification 183 (SessMgrRecoveryComplete)
Slot Number 9 Cpu Number 1 fetched from aaa mgr 1755 prior to audit 1755 passed
audit 1754 calls recovered 1754 all call lines 1754 time elapsed ms 1108.
Fri May 22 19:43:32 2015 Internal trap notification 1099 (ManagerRestart) facility
sessmgr instance 204 card 9 cpu 1
Fri May 22 19:44:49 2015 Internal trap notification 73 (ManagerFailure) facility
sessmgr instance 236 card 7 cpu 0
Fri May 22 19:44:49 2015 Internal trap notification 150 (TaskFailed) facility
sessmgr instance 236 on card 7 cpu 0
Fri May 22 19:44:49 2015 Internal trap notification 151 (TaskRestart) facility
sessmgr instance 236 on card 7 cpu 0
Fri May 22 19:44:51 2015 Internal trap notification 183 (SessMgrRecoveryComplete)
Slot Number 7 Cpu Number 0 fetched from aaa mgr 1741 prior to audit 1741 passed audit
1737 calls recovered 1737 all call lines 1737 time elapsed ms 1047.
Fri May 22 19:44:53 2015 Internal trap notification 1099 (ManagerRestart) facility
sessmgr instance 236 card 7 cpu 0
Fri May 22 19:50:04 2015 Internal trap notification 73 (ManagerFailure) facility
sessmgr instance 221 card 2 cpu 1
: Fri May 22 19:50:04 2015 Internal trap notification 150 (TaskFailed) facility
sessmgr instance 221 on card 2 cpu 1
Fri May 22 19:50:04 2015 Internal trap notification 151 (TaskRestart) facility
sessmgr instance 221 on card 2 cpu 1
Fri May 22 19:50:05 2015 Internal trap notification 183 (SessMgrRecoveryComplete)
Slot Number 2 Cpu Number 1 fetched from aaa mgr 1755 prior to audit 1755 passed
audit 1749 calls recovered 1750 all call lines 1750 time elapsed ms 1036.

***** show snmp trap history verbose *****

Fri May 22 19:43:10 2015 Internal trap notification 1099 (ManagerRestart) facility
sessmgr instance 204 card 9 cpu 1
Fri May 22 19:43:29 2015 Internal trap notification 73 (ManagerFailure) facility
sessmgr instance 204 card 9 cpu 1
Fri May 22 19:43:29 2015 Internal trap notification 150 (TaskFailed) facility
sessmgr instance 204 on card 9 cpu 1
Fri May 22 19:43:29 2015 Internal trap notification 151 (TaskRestart) facility
sessmgr instance 204 on card 9 cpu 1
Fri May 22 19:43:30 2015 Internal trap notification 183 (SessMgrRecoveryComplete)
Slot Number 9 Cpu Number 1 fetched from aaa mgr 1755 prior to audit 1755 passed
audit 1754 calls recovered 1754 all call lines 1754 time elapsed ms 1108.
Fri May 22 19:43:32 2015 Internal trap notification 1099 (ManagerRestart) facility
sessmgr instance 204 card 9 cpu 1
Fri May 22 19:44:49 2015 Internal trap notification 73 (ManagerFailure) facility
sessmgr instance 236 card 7 cpu 0
Fri May 22 19:44:49 2015 Internal trap notification 150 (TaskFailed) facility
sessmgr instance 236 on card 7 cpu 0
Fri May 22 19:44:49 2015 Internal trap notification 151 (TaskRestart) facility
sessmgr instance 236 on card 7 cpu 0
Fri May 22 19:44:51 2015 Internal trap notification 183 (SessMgrRecoveryComplete)
Slot Number 7 Cpu Number 0 fetched from aaa mgr 1741 prior to audit 1741 passed
audit 1737 calls recovered 1737 all call lines 1737 time elapsed ms 1047.
Fri May 22 19:44:53 2015 Internal trap notification 1099 (ManagerRestart) facility
sessmgr instance 236 card 7 cpu 0
Fri May 22 19:50:04 2015 Internal trap notification 73 (ManagerFailure) facility
sessmgr instance 221 card 2 cpu 1
: Fri May 22 19:50:04 2015 Internal trap notification 150 (TaskFailed) facility
sessmgr instance 221 on card 2 cpu 1
Fri May 22 19:50:04 2015 Internal trap notification 151 (TaskRestart) facility
sessmgr instance 221 on card 2 cpu 1
Fri May 22 19:50:05 2015 Internal trap notification 183 (SessMgrRecoveryComplete
) Slot Number 2 Cpu Number 1 fetched from aaa mgr 1755 prior to audit 1755 passed
audit 1749 calls recovered 1750 all call lines 1750 time elapsed ms 1036.

```
***** show logs *****
2015-May-25+23:15:53.123 [sitmain 4022 info] [3/1/4850 <sitmain:31> sittask.c:4762]
[software internal system critical-info syslog] Readdress requested for facility
sessmgr instance 5635 to instance 114
2015-May-25+23:15:53.122 [sitmain 4027 critical] [3/1/4850 <sitmain:31>
crash_mini.c:908] [software internal system callhome-crash] Process Crash Info:
time 2015-May-25+17:15:52(hex time 556358c8) card 03 cpu 01 pid 27118 procname
sessmgr crash_details
Assertion failure at acs/acsmgr/analyzer/ip/acs_ip_reasm.c:2970
Function: acsmgr_deallocate_ipv4_frag_chain_entry()
Expression: status == SN_STATUS_SUCCESS
Procllet: sessmgr (f=87000,i=114)
Process: card=3 cpu=1 arch=X pid=27118 cpu=~17% argv0=sessmgr
Crash time: 2015-May-25+17:15:52 UTC
Recent errno: 11 Resource temporarily unavailable
Stack (11032@0xfffffb000):
[ffffe430/X] __kernel_vsyscall() sp=0xffffbd28
[0af1delf/X] sn_assert() sp=0xffffbd68
[0891e137/X] acsmgr_deallocate_ipv4_frag_chain_entry() sp=0xffffbde8
[08952314/X] acsmgr_ip_frag_chain_destroy() sp=0xffffbee8
[089d87d1/X] acsmgr_process_tcp_packet() sp=0xffffc568
[089da270/X] acs_process_tcp_packet_normal_path() sp=0xffffc5b8
[089da3fd/X] acs_tcp_analyzer() sp=0xffffc638
[0892fb39/X] do_acsmgr_process_packet() sp=0xffffc668
[08940045/X] acs_ip_lean_path() sp=0xffffc6b8
[0887e309/X] acsmgr_data_receive_merge_mode() sp=0xffffc9d8
[0887f323/X] acs_handle_datapath_events_from_sm_interface() sp=0xffffca08
[037c2e1b/X] sessmgr_sef_initiate_data_packet_ind() sp=0xffffca88
[037c2f50/X] sessmgr_pcc_intf_send_data_packet_ind() sp=0xffffcaf8
[061de74a/X] sessmgr_pcc_fwd_packet() sp=0xffffcb58
[0627c6a4/X] sessmgr_ipv4_process_inet_pkt_part2_slow() sp=0xffffcf68
[06318343/X] sessmgr_ipv4_process_inet_pkt_pgw_ggsn() sp=0xffffd378
[0632196c/X] sessmgr_med_ipv4_data_received() sp=0xffffd418
[0633da9a/X] sessmgr_med_data_receive() sp=0xffffd598
[0afb977c/X] sn_epoll_run_events() sp=0xffffd5e8
[0afbdeb8/X] sn_loop_run() sp=0xffffda98
[0ad2b82d/X] main() sp=0xffffdb08

2015-May-25+23:15:53.067 [rct 13038 info] [5/0/7174 <rct:0> rct_task.c:305]
[software internal system critical-info syslog] Death notification of task
sessmgr/114 on 3/1 sent to parent task sessctrl/0 with failover of sessmgr/5635 on 3/1
2015-May-25+23:15:53.065 [evlog 2136 info] [5/0/7170 <evlogd:0> odule_persist.c:3102]
[software internal system critical-info syslog] Evlogd crashlog: Request received to
check the state of persistent crashlog.
2015-May-25+23:15:53.064 [sitmain 4099 info] [3/1/4850 <sitmain:31> crash_mini.c:765]
[software internal system critical-info syslog] have mini core, get evlogd status for
logging crash file 'crashdump-27118'
2015-May-25+23:15:53.064 [sitmain 4017 critical] [3/1/4850 <sitmain:31> sitproc.c:1544]
[software internal system syslog] Process sessmgr pid 27118 died on card 3 cpu 1
signal=6 wstatus=0x86
2015-May-25+23:15:53.048 [sitmain 4074 trace] [5/0/7168 <sitparent:50> crashd.c:1130]
[software internal system critical-info syslog] Crash handler file transfer starting
(type=2 size=0 child_ct=1 core_ct=1 pid=23021)
2015-May-25+23:15:53.047 [system 1001 error] [6/0/9727 <evlogd:1> evlgd_syslogd.c:221]
[software internal system syslog] CPU[3/1]: xmitcore[21648]: Core file transmitted to
card 5 size=663207936 elapsed=0sec:908ms
2015-May-25+23:15:53.047 [system 1001 error] [5/0/7170 <evlogd:0> evlgd_syslogd.c:221]
[software internal system syslog] CPU[3/1]: xmitcore[21648]: Core file transmitted to
card 5 size=663207936 elapsed=0sec:908ms
2015-May-25+23:15:53.047 [sitmain 4080 info] [5/0/7168 <sitparent:50> crashd.c:1091]
[software internal system critical-info syslog] Core file transfer to SPC complete,
received 8363207936/0 bytes
```

```
***** show session recovery status verbose *****
Tuesday May 26 05:55:26 BDT 2015
Session Recovery Status:
Overall Status : Ready For Recovery
Last Status Update : 8 seconds ago
```

```
-----sessmgr--- ----aaamgr---- demux
cpu state active standby active standby active status
-----
1/0 Active 24 1 24 1 0 Good
1/1 Active 24 1 24 1 0 Good
2/0 Active 24 1 24 1 0 Good
2/1 Active 24 1 24 1 0 Good
3/0 Active 24 1 24 1 0 Good
3/1 Active 24 1 24 1 0 Good
4/0 Active 24 1 24 1 0 Good
4/1 Active 24 1 24 1 0 Good
5/0 Active 0 0 0 0 14 Good (Demux)
7/0 Active 24 1 24 1 0 Good
7/1 Active 24 1 24 1 0 Good
8/0 Active 24 1 24 1 0 Good
8/1 Active 24 1 24 1 0 Good
9/0 Active 24 1 24 1 0 Good
9/1 Active 24 1 24 1 0 Good
10/0 Standby 0 24 0 24 0 Good
10/1 Standby 0 24 0 24 0 Good
```

충돌 로깅 아키텍처

크래시 로그는 소프트웨어 크래시(전체 코어 덤프)와 관련된 모든 가능한 정보를 기록합니다. 크기 때문에 시스템 메모리에 저장할 수 없습니다.따라서 이러한 로그는 시스템이 로컬 디바이스를 가리키는 URL로 구성되거나 로그가 저장될 수 있는 네트워크 서버를 가리키는 경우에만 생성됩니다.

충돌 로그는 충돌 이벤트 정보의 영구 저장소입니다.각 이벤트는 번호가 지정되며 CPU(minicore), NPU(network processing unit) 또는 커널 충돌과 관련된 텍스트가 포함됩니다.로깅된 이벤트는 고정 길이 레코드에 기록되고 /flash/crashlog2에 저장됩니다.

충돌이 발생할 때마다 이 충돌 정보가 저장됩니다.

1. 이벤트 레코드는 /flash/crashlog2 파일(충돌 로그)에 저장됩니다.
2. 연결된 미니코어, NPU 또는 커널 덤프 파일은 /flash/crsh2 디렉토리에 저장됩니다.
3. 전체 코어 덤프는 사용자 구성 디렉토리에 저장됩니다.

관리 카드 간 충돌 이벤트 및 미니코어 동기화

crashlog는 각 관리 카드에 고유하므로 카드 "8"이 활성화될 때 충돌이 발생하면 카드 "8"에 기록됩니다.후속 전환은 더 이상 로그에 크래시를 표시하지 않습니다.이 충돌을 검색하려면 "8" 카드로 다시 전환해야 합니다.충돌 이벤트 로그 및 덤프는 활성 및 대기 관리 카드에 고유하므로 활성 카드에 충돌이 발생하면 충돌 이벤트 로그 및 관련 덤프는 활성 카드에만 저장됩니다.이 충돌 정보는 스탠바이 카드에서 사용할 수 없습니다.활성 카드의 충돌로 인해 카드가 전환되고 장애 정보가 더 이상 카드에 표시되지 않을 때마다 현재 활성 카드에서만 충돌 정보를 검색할 수 있습니다.다른 카드의 충돌 목록을 검색하려면 스위치오버가 다시 필요합니다.이러한 전환을 방지하고 스탠바이 카드에서 충돌 정보를 얻으려면 두 관리 카드 간의 동기화와 최신 충돌 정보의 유지 관리가 필요합니다.

도착하는 충돌 이벤트는 대기 SMC/MIO로 전송되고 대기자의 crashlog 파일에 비슷한 방식으로 저

장됩니다. 활성 SMC/MIO의 플래시에 있는 Minicore, NPU 또는 커널 덤프는 rsync 명령을 사용하여 대기 SMC/MMIO에 동기화해야 합니다. crashlog 항목 또는 전체 목록이 CLI 명령을 통해 삭제될 경우 활성 및 대기 SMCs/MIO에서 지워져야 합니다. 메모리에 영향을 주지 않습니다. 대기 이벤트 로드가 적고 대기 카드에 동기화 작업을 위한 공간이 충분하므로 모든 충돌 관련 동기화 작업은 대기 SMC/MIO 카드의 이벤트 로그에 의해 수행됩니다. 따라서 시스템의 성능에 영향을 미치지 않습니다.

명령

다음 명령을 사용하여 문제를 해결할 수 있습니다.

```
#show support details
```

```
#show crash list
```

```
#show logs
```

```
#show snmp trap history verbose
```

```
#show session recovery status verbose
```

```
#show task resources facility sessmgr instance <>
```

```
#show task resources facility sessmgr all
```

코어파일은 충돌 후 생성됩니다. 일반적으로 운영자는 외부 서버에 저장합니다. 코어파일 이름은 일반적으로 crash-<Cardnum>-<CPU Num>-<Hex timestamp>-core.gcrash-09-00-5593a1b8-core와 같습니다.

충돌이 발생할 때마다 이 충돌 정보가 저장됩니다.

- 이벤트 레코드는 /flash/crashlog2 파일(충돌 로그)에 저장됩니다.
- 연결된 미니코어, NPU 또는 커널 덤프 파일은 /flash/crsh2 디렉토리에 저장됩니다.

요약

모든 ASR5x00 소프트웨어는 예상되는 조건/이벤트와 예기치 않은 상황/이벤트를 모두 처리하도록 설계되었습니다. Cisco는 완벽한 소프트웨어를 얻기 위해 노력하지만, 실수가 불가피하며 충돌이 발생할 수 있습니다. 따라서 세션 복구 기능이 매우 중요합니다. Cisco의 완벽한 노력은 충돌 발생을 최소화하며, 세션 복구는 충돌 후 세션을 계속할 수 있게 합니다. 그럼에도 불구하고 Cisco는 완벽한 소프트웨어를 얻기 위해 계속 노력하는 것이 중요합니다. 충돌 수가 적을수록 동시에 발생하는 여러 충돌 가능성이 줄어듭니다. 세션 복구는 단일 충돌을 원활하게 해결하지만 여러 동시 충돌 시 발생하는 복구는 약간 다르게 설계되었습니다. 운영자는 여러 건의 동시 충돌을 거의 발생(또는 발생 안 함)하지 않아야 하지만, 이러한 사고가 발생할 경우 ASR5x00은 일부 가입자 세션의 희생으로 시스템 무결성을 가장 우선순위로 복구하도록 설계되었습니다.