ASR5x00 sessiebeheertaken - beschrijving van de functies, crashes, herstelbewerkingen en vastlegging

Inhoud

Inleiding

Dit document beschrijft en verklaart de betrouwbaarheid van de software, de beschikbaarheid van de service en de functies voor failover voor Cisco Aggregation Services Router (ASR) 5x00 Series. Het geeft de definitie van een softwarecrash bij ASR5x00 en de effecten van de softwarecrash. Het artikel gaat verder om vast te stellen dat zelfs in het geval van onverwachte softwarecrashes, hoe de ASR5x00 het doel van beschikbaarheid van de "carrièreklasse" kan bereiken dankzij de inherente software-veerkracht en de beschikbaarheid van functies. De mobiele abonnee zou nooit moeten nadenken over de beschikbaarheid van de service. Het doel van Cisco is geen sessieverlies door één hardware- of softwarefout, zoals het verlies van een compleet systeem, met andere woorden - betrouwbaarheid van spraakkwaliteit. De functies voor de betrouwbaarheid van de software op ASR5x00 zijn bedoeld om de doelstellingen voor de beschikbaarheid van de dienst van "carrierklasse" te bereiken, zelfs in gevallen waarin zich onvoorziene storingen in het netwerk van een exploitant kunnen voordoen.

Softwarearchitectuur: Ontworpen voor veerkracht

ASR5x00 heeft een verzameling softwaretaken die worden verspreid over de Packet Services Card (PSC) of Data Processing Card (DPC) en System Management Card (SMC) of Management and I/O (MIO) kaarten die zijn ontworpen om een verscheidenheid aan specifieke functies te vervullen.

De taak van de sessiebeheerder is bijvoorbeeld verantwoordelijk voor de verwerking van de sessies van een verzameling abonnees en voor het uitvoeren van inline services zoals peer-to-peer (P2P), Deep Packet Inspection (DPI) en dergelijke op gebruikersverkeer. De taak van de

beheerder Verificatie, autorisatie en accounting (AAA) is verantwoordelijk voor het genereren van factureringsgebeurtenissen om abonneeverkeersgebruik enzovoort op te nemen. De sessiemanager en AAA-beheertaken worden uitgevoerd op de PSC/DPC-kaart.

De SMC/MIO-kaart is voorbehouden voor exploitatie en onderhoud (O&M) en aan het platform gerelateerde taken. Het ASR5x00-systeem is virtueel gecompartimenteerd in verschillende software-subsystemen, zoals het sessie-subsysteem voor het verwerken van abonnementssessies en het VPN-subsysteem dat verantwoordelijk is voor de IP-adrestoewijzing, -routing enzovoort. Elk subsysteem heeft een controletaak die toezicht houdt op de gezondheid van het subsysteem dat het controleert. De controllertaken worden uitgevoerd op de SMC/MIO-kaart. De sessiebeheer- en AAA-beheertaken worden samengevoegd om de sessie van de abonnee af te handelen voor controle-, gegevensverkeer- en factureringsdoeleinden. Wanneer de sessieherstel in het systeem is ingeschakeld, maakt elke sessiemanager een back-up van de status van de abonneestaten met een peer AAA-beheertaak die moet worden hersteld wanneer een sessie manager crasht.

Wat is een ongeluk?

Een taak in de ASR5x00 kan mogelijk crashen als er tijdens normaal gebruik een storing optreedt. Een storing of softwarefout in de ASR5x00 is gedefinieerd als een *onverwachte* uitgang of beëindiging van een taak in het systeem. Een crash kan plaatsvinden als de software code probeert om toegang te krijgen tot verboden geheugengebieden (zoals beschadigde gegevensstructuren), een toestand in de code tegenkomt die niet verwacht wordt (zoals een ongeldige staatsovergang), enzovoort. Een crash kan ook worden geactiveerd als de taak niet meer reageert op de systeemmonitor taak en de monitorpogingen om de taak te doden en opnieuw te starten. Een crash-event kan ook expliciet worden geactiveerd (in tegenstelling tot verwacht) in het systeem wanneer een taak gedwongen wordt om de huidige status te dumpen door een CLI-opdracht of door de systeemmonitor om de taakstaat te analyseren. Een verwachte crash-gebeurtenis kan ook plaatsvinden wanneer de systeemcontroller-taken zichzelf opnieuw opstarten om een situatie te corrigeren met een beheertaak die herhaaldelijk faalt.

Gevolgen van een ongeluk met een sessiebeheer

Bij normaal gebruik beheert een sessiebeheerder een verzameling abonneesessies en gekoppeld gegevensverkeer voor de sessies samen met een baanbrekende AAA Manager-taak die het factureren voor deze abonneesessies regelt. Wanneer een sessie manager crash optreedt, bestaat het niet meer in het systeem. Als de sessieherstel in het systeem is ingeschakeld, wordt er een stand-by sessiemanager-taak uitgevoerd om actief te worden op dezelfde PSC/DPC-kaart. Deze nieuwe sessie Manager taak herstelt de abonneesessies terwijl deze communiceert met de peer AAA manager-taak. De terugwinningshandeling varieert van 50 msec tot enkele seconden afhankelijk van het aantal sessies dat actief was in de sessiemanager op het moment van de crash en de totale CPU-lading op de kaart, enzovoort. Er is geen verlies in abonnementssessies die reeds in de oorspronkelijke sessiemanager bij deze operatie waren ingesteld. Alle abonnees die op het moment van de crash in het leven werden geroepen, zullen waarschijnlijk ook worden hersteld door middel van protocol-terugzendingen enzovoort. Alle datapakketten die op het moment van de crash in het systeem waren doorgevoerd, kunnen verondersteld worden te zijn gekoppeld aan een netwerkverlies door de communicerende entiteiten van de netwerkverbinding. Ze zullen opnieuw worden verzonden en de verbinding zal worden uitgevoerd door de nieuwe sessiebeheerder. De factureringsinformatie voor de sessies die door de sessieleider worden

gedragen, wordt bewaard in de peer AAA-manager.

Wanneer moet de exploitant zich zorgen maken?

Wanneer een crash van de sessiemanager plaatsvindt, gebeurt de herstelprocedure zoals eerder beschreven en de rest van het systeem blijft door deze gebeurtenis onaangetast. Een crash in één sessiemanager heeft geen invloed op de andere sessiemanagers. Als richtlijn voor de exploitant, indien meerdere sessiemanagers taken *op dezelfde PSC/DPC kaart* gelijktijdig of binnen 10 minuten van elkaar uitvoeren, kunnen er sessies verloren gaan omdat het systeem niet in staat zou zijn om snel genoeg nieuwe sessiemanagers te starten om de plaats van de verongelukte taken in te nemen. Dit komt overeen met een dubbel foutscenario waarin het verlies van sessies kan voorkomen. Als herstel niet mogelijk is, wordt de sessieleider simpelweg opnieuw opgestart en is hij bereid nieuwe sessies te accepteren.

Wanneer een bepaalde sessiemanager herhaaldelijk crasht (bijvoorbeeld wanneer meerdere malen dezelfde fout optreedt), neemt de sessie-controller de taak op en start deze opnieuw in een poging het subsysteem te herstellen. Indien de taak van de sessiecontrole het subsysteem van de sessie niet kan stabiliseren en zichzelf bij deze inspanning voortdurend opnieuw begint, is de volgende stap in de escalatie dat het systeem naar een stand-by SMC/MIO-kaart moet switches. In het onwaarschijnlijke geval dat er geen stand-by SMC/MIO-kaart is of indien er een storing optreedt in de overschakelingswerking, herstart het systeem zichzelf.

Session managers houden ook statistieken bij voor elk Access Point Name (APN), Services, functies, enzovoort, die permanent verloren zullen worden wanneer een crash plaatsvindt. Daarom zal een externe entiteit die lampen verzamelt periodiek een dip in de statistieken waarnemen wanneer een of meer crashes optreden. Dit kan zich manifesteren als een dip in een grafische voorstelling van de statistieken die over een tijdas worden getrokken.

> **Opmerking:** Een typisch chassis met 7-14 PSC- of 4-10 DPC-kaarten heeft ongeveer 120-160 sessiemanagers, afhankelijk van het aantal PSC/DPC-kaarten, en één enkele crash zal het verlies van ongeveer 1/40^{e} of 1/80^{e} van de statistieken tot gevolg hebben. Als een stand-by sessiemanager de controle overneemt, zal hij de statistieken vanaf nul opnieuw verzamelen.

Hoe weten we of er een ongeluk is gebeurd?

Een crash zal een SNMP-trap-gebeurtenis activeren naar een netwerk monitoring station, zoals de Event Monitoring Service (EMS) en door syslog-gebeurtenissen. De crashes die in het systeem zijn voorgekomen, kunnen ook worden waargenomen met de opdracht **showcrashlist**. Merk op dat deze opdracht zowel onverwachte als verwachte crashgebeurtenissen aantoont zoals eerder beschreven. Deze twee soorten crashgebeurtenissen kunnen worden onderscheiden door middel van een header die elke crash beschrijft.

Een taak die wordt gevolgd door een succesvol herstel van de sessie wordt aangegeven door dit logbericht:

```
"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#>"
```

Een taak die niet kon herstellen wordt door dit logbericht aangegeven:

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

Samengevat: als herstel van de sessie is ingeschakeld, worden de crashes in de meeste gevallen niet opgemerkt omdat ze geen impact hebben van de abonnee. Je moet de CLI-opdracht invoeren of de logbestanden of SNMP-waarschuwing bekijken om elk voorval van crashes te kunnen detecteren.

Bijvoorbeeld:

```
 ******** 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
Proclet: 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@0xffffb000):
[ffffe430/X] __kernel_vsyscall() sp=0xffffbd28
[0af1de1f/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
```

Crash Logging Architecture

Crash logs nemen alle mogelijke informatie op die betrekking heeft op een softwarecrash (full-core-dumpen). Vanwege hun grootte kunnen ze niet in het systeemgeheugen worden opgeslagen. Daarom worden deze logbestanden alleen gegenereerd als het systeem is geconfigureerd met een URL die naar een lokaal apparaat of een netwerkserver wijst waar het logbestand kan worden opgeslagen.

Het crashlogboek is een persistente opslagplaats van informatie over crashgebeurtenissen. Elke gebeurtenis is genummerd en bevat tekst die is gekoppeld aan een CPU (minicore), een netwerkverwerkingseenheid (NPU) of een kernongeluk. De ingelogde gebeurtenissen worden vastgelegd in vaste lengte records en opgeslagen in /flitser/crashlog2.

Wanneer een crash optreedt, wordt deze crashinformatie opgeslagen:

1. Het eventrecord wordt opgeslagen in /flash/crashlog2 bestand (het crashlogbestand).
2. Het gekoppelde minicore-, NPU- of kernel-dumpbestand wordt opgeslagen in de /flash/crsh2-map.
3. Een volle kern-stortplaats wordt opgeslagen in een door gebruiker gevormde folder.

Synchronisatie van crashgebeurtenissen en minicores tussen beheerkaarten

De crashlog is uniek voor elk van de beheerkaarten, dus als er een crash optreedt wanneer kaart "8" actief is, zal deze worden ingelogd op kaart "8". Een volgende omschakeling zou niet langer de krach in het logboek tonen. Om dit ongeluk terug te krijgen, moet een switch naar kaart "8" worden teruggestuurd. Het crashevenement logbestand en de dumps zijn uniek voor actieve en stand-by beheerkaarten. Als er een crash op een actieve kaart plaatsvindt, worden het crashevenement en de bijbehorende dumps alleen op een actieve kaart opgeslagen. Deze crashinformatie is niet beschikbaar op de stand-by kaart. Wanneer de kaartoverschakeling als gevolg van een crash in de actieve kaart, en de crashinformatie niet langer op de kaart wordt weergegeven die overneemt, kan de crashinformatie alleen van de huidige actieve kaart worden opgeroepen. Om de crashlijst van de andere kaart weer op te halen is een omschakeling opnieuw vereist. Om deze omschakeling te vermijden en de crashinformatie van de standby kaart te verkrijgen, is synchronisatie tussen twee beheerkaarten en onderhoud van de laatste crashinformatie nodig.

De aankomende gebeurtenis zal naar de stand-by SMC/MIO worden verzonden en op de zelfde manier in het crashlogbestand van de standby worden opgeslagen. Minicore, NPU, of kernel dumps op actieve SMC/MIO-flitser moeten met de **sync-**opdracht gesynchroniseerd worden naar stand-by SMC/MMIO. Wanneer een crashlogingang of de hele lijst door de CLI-opdracht wordt verwijderd, moet deze op zowel actieve als standby SMCs/MIOs worden gewist. Er is geen invloed op het geheugen. Alle synchronisatie-activiteit die met een crash te maken heeft, wordt uitgevoerd door het evlogd van de stand-by SMC/MIO-kaart, aangezien het standby-evlogd minder geladen is en de stand-by kaart voldoende ruimte voor synchronisatieactiviteit heeft. De prestaties van het systeem zullen derhalve niet worden beïnvloed.

Opdrachten

Deze opdrachten kunnen worden gebruikt om problemen met de oplossing op te lossen:

```
#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
```

Corefiles worden gegenereerd na een ongeluk. Gewoonlijk slaan de operatoren ze op een externe server op. De corefile naam lijkt meestal op crashes-<Cardnum>-<CPU Num>-<Hex timestamp>-coree.gcrasht-09-00-5593a1b8-core.

Wanneer een crash optreedt, wordt deze crashinformatie opgeslagen:

- Het eventrecord wordt opgeslagen in /flash/crashlog2 bestand (het crashlogbestand).
- Het gekoppelde minicore-, NPU- of kernel-dumpbestand wordt opgeslagen in de /flash/crsh2-map.

Samenvatting

Alle ASR5x00-software is ontworpen om zowel de voorziene omstandigheden/gebeurtenissen als de onvoorziene omstandigheden/gebeurtenissen aan te kunnen. Terwijl Cisco ernaar streeft perfecte software te hebben, zullen onvermijdelijk fouten bestaan en crashes mogelijk zijn. Dat is de reden dat de sessie recovery optie zo belangrijk is. De poging van Cisco om de perfectie te perfectioneren zal het voorkomen van crashes minimaliseren, en de sessieherstel zal de sessies toestaan om na een crash verder te gaan. Niettemin is het belangrijk dat Cisco blijft streven naar perfecte software. Minder crashes verminderen de kans op meerdere crashes die tegelijkertijd plaatsvinden. Terwijl het herstel van de sessie naadloos één crash geneest, wordt het herstel van meerdere gelijktijdige crashes een beetje anders ontworpen. Exploitanten dienen zelden (of nooit) meerdere gelijktijdige crashes te ervaren, maar als dit gebeurt is de ASR5x00 ontworpen om de systeemintegriteit als hoogste prioriteit terug te krijgen, mogelijk na opoffering van een aantal abonnementssessies.