

ASR5x00會話管理器任務 — 功能、崩潰、恢復操作和崩潰日誌的說明

目錄

[簡介](#)

[軟體體系結構：專為恢復能力而設計](#)

[什麼是崩潰？](#)

[會話管理器崩潰的影響](#)

[操作員何時應感到擔心？](#)

[如何知道是否發生了崩潰？](#)

[崩潰記錄體系結構](#)

[在管理卡之間同步崩潰事件和微核心](#)

[指令](#)

[摘要](#)

簡介

本文檔介紹並說明思科聚合服務路由器(ASR)5x00系列的軟體可靠性、服務可用性和故障轉移功能。它介紹了ASR5x00上軟體崩潰的定義以及軟體崩潰的影響。文章進一步指出，即使在出現意外軟體崩潰的情況下，ASR5x00也能通過固有的軟體恢復能力和可用性功能實現「運營商級」可用性目標。移動使用者永遠不必考慮服務的可用性。思科的目標是不會因為任何單個硬體或軟體故障而丟失會話，這些故障包括丟失整個系統，換句話說，即語音級別的可靠性。ASR5x00上的軟體可靠性功能旨在實現「運營商級」服務可用性的目標，即使在運營商網路中可能出現不可預見的故障時也是如此。

軟體體系結構：專為恢復能力而設計

ASR5x00具有分佈在資料包服務卡(PSC)或資料處理卡(DPC)以及系統管理卡(SMC)或管理和I/O(MIO)卡上的一系列軟體任務，這些任務旨在執行各種特定功能。

例如，會話管理器任務負責處理一組使用者的會話並對使用者流量執行內聯服務，例如對等(P2P)、深度資料包檢測(DPI)等。身份驗證、授權和記帳(AAA)管理器任務負責生成計費事件，以記錄使用者流量使用率等。會話管理器和AAA管理器任務在PSC/DPC卡上運行。

SMC/MIO卡保留用於操作與維護(O&M)和平台相關任務。ASR5x00系統實際上被劃分為不同的軟體子系統，例如用於處理使用者會話的會話子系統，以及負責IP地址分配、路由等的VPN子系統。每個子系統都有一個控制器任務，用於監視其控制的子系統的運行狀況。控制器任務在SMC/MIO卡上運行。會話管理器和AAA管理器任務配對在一起，以便處理使用者的會話以進行控制、資料流量和計費。在系統中啟用會話恢復後，每個會話管理器任務都會備份其使用者狀態集的狀態，並在會話管理器崩潰時備份要恢復的對等AAA管理器任務。

什麼是崩潰？

如果ASR5x00中的任務在正常運行期間遇到故障情況，則它可能會崩潰。ASR5x00中的故障或軟體故障被定義為系統中任務的意外退出或終止。如果軟體代碼嘗試訪問被禁止的記憶體區域（如損壞的資料結構），遇到代碼中不應有的條件（如無效狀態轉換），等等，則會發生崩潰。如果任務對系統監控器任務變得無響應，並且監控器嘗試終止和重新啟動該任務，則也會觸發崩潰。當任務被CLI命令強制轉儲其當前狀態時，系統監視器也可以顯式觸發崩潰事件（而非意外事件），以便分析任務狀態。當系統控制器任務重新啟動時，也可能發生預期的故障事件，以便有可能更正多次失敗的管理器任務的情況。

會話管理器崩潰的影響

在正常操作下，會話管理器任務處理一組使用者會話和會話的相關資料流量，以及處理這些使用者會話計費的對等AAA管理器任務。當會話管理器發生崩潰時，它將不再存在於系統中。如果在系統中啟用會話恢復，則備用會話管理器任務將在同一PSC/DPC卡中變為活動狀態。此新會話管理器任務在與對等AAA管理器任務通訊時恢復使用者會話。恢復操作從50毫秒到幾秒不等，具體取決於崩潰時會話管理器中活動的會話數以及卡上的總CPU負載等。在此操作中，已在原始會話管理器中建立的使用者會話不會丟失。由於協定重新傳輸等原因，崩潰時正在建立的任何使用者會話也可能會恢復。在崩潰時通過系統轉換的任何資料包都可以被網路連線的通訊實體假定為與網路丟失相關聯，並且將被重新傳輸並且連線將由新的會話管理器繼續。會話管理器所承載會話的計費資訊將保留在對等AAA管理器中。

操作員何時應感到擔心？

當會話管理器崩潰時，恢復過程按前面所述進行，而系統的其餘部分不受此事件的影響。一個會話管理器中的崩潰不會影響其他會話管理器。作為操作員的指導，如果同一PSC/DPC卡上的多個會話管理器任務同時崩潰或相隔10分鐘內，則可能會丟失會話，因為系統可能無法以足夠快的速度啟動新的會話管理器來代替崩潰的任務。這對應於可能出現會話丟失的雙重故障情況。當恢復不可行時，會話管理器只需重新啟動，即可接受新的會話。

當給定的會話管理器反復崩潰（例如反復遇到相同的故障條件）時，會話控制器任務會記下並重新啟動自己，以嘗試恢復子系統。如果會話控制器任務無法穩定會話子系統，並在此過程中不斷重新啟動自身，則升級中的下一步是系統切換到備用SMC/MIO卡。如果沒有待命SMC/MIO卡，或者切換操作中遇到故障，這種可能性很小，在這種情況下，系統自行重新啟動。

會話管理器還維護每個接入點名稱(APN)、服務、功能等的統計資訊，在發生崩潰時將永久丟失這些統計資訊。因此，定期收集批次統計資料的外部實體將在發生一個或多個崩潰時觀察到統計資訊下降。這可以表示為在時間軸上繪製的統計資料的圖形表示中的凹陷。

附註：使用7-14個PSC或4-10個DPC卡填充的典型機箱約有120-160個會話管理器，這取決於PSC/DPC卡的數量，單個故障會導致丟失統計資訊的 $1/40$ 或 $1/80$ 。當備用會話管理器接管時，它開始再次從零開始累積統計資訊。

如何知道是否發生了崩潰？

崩潰會向網路監控站觸發SNMP陷阱事件，例如事件監控服務(EMS)和系統日誌事件。也可使用

show crash list命令觀察系統中發生的崩潰。請注意，此命令同時列出意外和預期的故障事件，如前所述。這兩種型別的崩潰事件可以通過描述每個崩潰的報頭來區分。

以下日誌消息表明發生了任務崩潰，隨後會話恢復成功：

```
"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#>"
```

以下日誌消息指示無法恢復的任務崩潰：

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

總而言之，啟用會話恢復後，大多數情況下不會發現崩潰，因為它們對使用者沒有影響。必須輸入CLI命令，或檢視日誌或SNMP通知，才能檢測任何發生的崩潰。

例如：

```
***** 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=0xfffffbd28
[0af1delf/X] sn_assert() sp=0xfffffbd68
[0891e137/X] acsmgr_deallocate_ipv4_frag_chain_entry() sp=0xfffffbd28
[08952314/X] acsmgr_ip_frag_chain_destroy() sp=0xfffffbee8
```

```
[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
```

崩潰記錄體系結構

崩潰日誌記錄與軟體崩潰（完全核心轉儲）相關的所有可能資訊。由於其大小，它們無法儲存在系統記憶體中。因此，僅當系統配置了指向可以儲存日誌的本地裝置或網路伺服器的URL時，才會生成這些日誌。

崩潰日誌是崩潰事件資訊的永久儲存庫。每個事件都進行編號，並包含與CPU(minicore)、網路處理單元(NPU)或核心崩潰關聯的文本。記錄的事件記錄到固定長度的記錄中，並儲存在 /flash/crashlog2 中。

每當發生崩潰時，都會儲存此崩潰資訊：

1. 事件記錄儲存在 /flash/crashlog2 檔案中（故障日誌）。
2. 關聯的minicore、NPU或核心轉儲檔案儲存在 /flash/crsh2 目錄中。
3. 完全核心轉儲儲存在使用者配置的目錄中。

在管理卡之間同步崩潰事件和微核心

崩潰日誌是每個管理卡的唯一資訊，因此，如果卡「8」處於活動狀態時發生崩潰，它將登入到卡「8」。隨後的切換操作將不再在日誌中顯示崩潰。為了擷取此崩潰訊息，必須重新切換到卡「8」。崩潰事件日誌和轉儲對於主用和備用管理卡是唯一的，因此，如果在主用卡上發生崩潰，則崩潰事件日誌和相關轉儲將只儲存在主用卡上。此崩潰資訊在備用卡上不可用。每當卡由於活動卡中的故障而切換，且故障資訊不再顯示在接管卡上，只能從當前活動卡檢索故障資訊。為了檢索另一卡的故障清單，需要再次進行切換。為了避免這種切換，並從備用卡獲取故障資訊，需要在兩個管理卡之間進行同步並維護最新的故障資訊。

到達的故障事件將傳送到備用SMC/MIO，並以類似方式儲存在備用的crashlog檔案中。活動SMC/MIO的快閃記憶體上的Minicore、NPU或核心轉儲需要使用rsync命令同步到備用SMC/MMIO。通過CLI命令刪除崩潰日誌條目或整個清單時，應在主用和備用SMC/MIO上清除該條目。對記憶體沒有影響。所有與故障相關的同步活動都將由備用SMC/MIO卡的evlogd完成，因為備用evlogd的載入較少，並且備用卡有足夠的空間用於同步活動。因此不會影響系統的效能。

指令

以下命令可用於排除故障：

```
#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
```

崩潰後生成核心檔案。通常操作員將它們儲存在外部伺服器上。核心檔名稱通常看起來像是crash-
<Cardnum>-<CPU Num>-<Hex timestamp>-coree.gcrash-09-00-5593a1b8-core。

每當發生崩潰時，都會儲存此崩潰資訊：

- 事件記錄儲存在/flash/crashlog2檔案中（故障日誌）。
- 關聯的minicore、NPU或核心轉儲檔案儲存在/flash/crsh2目錄中。

摘要

所有ASR5x00軟體都設計為可同時處理可預見的條件/事件和不可預見的條件/事件。儘管思科努力擁有完美的軟體，但不可避免的錯誤會存在，並且崩潰是可能的。這就是會話恢復功能如此重要的原因。思科的完美工作將使崩潰發生率降至最低，會話恢復將允許會話在崩潰後繼續運行。儘管如此，思科繼續致力於實現完美的軟體仍很重要。更少的崩潰次數將降低同時發生多個崩潰的可能性。雖然會話恢復可以無縫地修復單個崩潰，但多個同時崩潰的恢復設計卻略有不同。運營商很少會（或永遠不會）發生多次同時崩潰，但是如果發生這種情況，ASR5x00旨在恢復系統完整性作為最高優先順序，可能犧牲一些使用者會話。