

ASR5x00会话管理器任务-功能、失败、恢复操作和失败日志的说明

目录

[简介](#)

[软件体系结构：设计为弹性](#)

[什么是失败？](#)

[会话管理器失败的作用](#)

[操作员什么时候应该获得关心？](#)

[如何知道失败是否出现？](#)

[失败记录日志体系结构](#)

[失败事件和Minicores的同步在管理卡之间](#)

[命令](#)

[摘要](#)

简介

本文描述并且解释软件可靠性、提供的服务和故障切换功能5x00系列Cisco聚合服务的路由器的(ASR)。它提交一个软件崩溃的定义在ASR5x00和软件崩溃的作用。条款设立那在意外的软件崩溃的情况下，ASR5x00如何能提供“载波类”可用性由于内在的软件弹性和可用性目标功能。移动用户不应该必须考虑服务的可用性。思科的目标是没有会话丢失由于所有单个硬件或软件故障，包括完整系统的损耗，换句话说-音频级的可靠性。在未预见到的失败在操作员的网络处，也许发生在ASR5x00的软件可靠性功能被瞄准能达到“均等载波类”的提供的服务的目标。

软件体系结构：设计为弹性

ASR5x00有软件任务的一集在设计执行各种各样的特定功能的信息包服务卡德(PSC)或数据处理卡德(DPC)和系统管理卡德(SMC)或管理和I/O (减少)卡间被分配的。

例如，会话管理器任务对处理一套的会话负责用户和进行轴向服务例如对等(P2P)，深度信息包检验(DPI)，等等，在用户数据流。验证、授权和统计(AAA)管理器任务对计费事件的生成负责为了记录用户数据流使用情况等等。在PSC/DPC卡运行的会话管理器和AAA管理器任务。

SMC/MIO卡为运行和维护(肿瘤)和平台涉及的任务保留。ASR5x00系统实际上被分区到不同的软件子系统例如处理的用户会话子系统和对IP地址分配负责，路由，等等的VPN子系统。每个子系统有监督子系统健康控制的一控制器任务。在SMC/MIO卡运行的控制器任务。任务一起配对会话管理器和AAA管理器为了处理控制、数据流和计费目的用户的会话。当会话恢复在系统启用，在会话管理器失败情形下，每会话管理器任务备份其套的状态有对等体AAA管理器任务的用户状态被恢复。

。

什么是失败？

如果在正常操作时，遇到故障状况在ASR5x00的一任务能潜在失败。在ASR5x00的失败或软件故障定义是一任务的一个意外的退出或终端在系统的。失败能发生，如果软件代码尝试到被禁止的访问存储器内存区(例如损坏的数据结构)，遇到在没有预计的代码的一个条件(例如一个无效状态转换)，等等。如果任务变得无答复对系统显示器任务和监视器尝试杀害和重新启动任务，失败可能也被触发。失败事件可能明确地也触发(与意外相对)在系统，当任务由CLI命令时被迫转存其当前状态或由系统显示器为了分析任务状态。一个预计失败事件能也发生，当潜在正确系统控制器任务的重新启动为了与重复发生故障的管理器任务的一个情况。

会话管理器失败的作用

在正常操作下，会话管理器任务与该一并列的AAA管理器的任务一起处理一套用户会话和相关的数数据流会话的发单为那些用户会话的把柄。当会话管理器失败出现时在系统停止存在。如果会话恢复在系统启用，一暂挂会话管理器任务做变得激活在同一个PSC/DPC卡。当与对等体AAA管理器任务，联络此个新会话管理器任务复原用户会话。恢复操作范围自50毫秒到一些秒钟从属在是活跃的在会话管理器在失败时和在卡的整体CPU负载等等会话的数量。没有损耗在此操作的原始会话管理器已经建立的用户会话上。是在建立过程中在失败时的所有用户会话可能也恢复的归结于协议重新传输等等。在转换通过系统在失败时能假设关联与网络损耗由网络连接通信实体，并且被重新传输的所有数据包和连接将由个新会话管理器坚持。会话管理器传送的会话的计费信息在对等体AAA管理器将保留。

操作员什么时候应该获得关心？

当会话管理器失败出现时，恢复流程发生如以前描述和系统的其余依然是未受影响由此事件。在一会话管理器的失败不影响其他会话管理器。作为对操作员的一个指导，如果多个会话管理器任务在**同一PSC/DPC卡失败同时或在10分钟彼此内**，那里也许在会话损耗，因为系统也许不能开始足够快速个新会话的管理器代替失败的任务。这对应于会话损耗能发生的一个双故障方案。当恢复不可行时，会话管理器重新启动并且准备接受新建的会话。

当给的会话管理器重复失败(例如它多次遇到同一个故障状况)，会话控制器任务注意并且重新启动为恢复子系统。如果会话控制器任务无法稳定会话子系统并且重新启动不断地在此努力，在逐步升级的下一步是为了系统能转换到待机SMC/MIO卡。在不太可能的情况没有待机SMC/MIO卡或，如果失败在切换操作遇到，系统重新启动。

会话管理器也维护每接入点名称的(APN)统计信息，服务，将永久丢失的functionalites，等等，当失败出现时。所以收集bulkstats的外部实体周期地将观察在统计信息的一个垂度，当一个或更多失败出现。这能表明作为在统计信息的图形表示的一个垂度被画时间轴。

注意：一个典型的机箱带有7-14 PSC或4-10个DPC卡有关于120-160会话管理器，从属在PSC/DPC卡数量，并且单个失败将导致大约1/40Th或统计信息的1/80Th损耗。当暂挂会话管理器接管时，它开始再累计统计信息从零。

如何知道失败是否出现？

失败触发SNMP陷阱事件对网络监控控制台，例如事件监控服务(EMS)和将由系统日志事件。在系

统发生的失败可能也观察与list命令显示的失败。注意此命令一览表意外和预计失败事件如描述前。这两个故障类型事件可以是著名的通过描述每失败的报头。

成功的会话恢复跟随的任务失败是由此日志消息表示的：

```
"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#>"
```

不可能恢复的任务失败是由此日志消息表示的：

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

总之，因为他们没有用户影响，当会话恢复启用，在大多数情况下失败不会被注意。一必须输入CLI命令，或者请查看日志或SNMP通知为了检测失败所有出现。

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@0xffffb000):
[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
```

失败记录日志体系结构

适合于对软件崩溃的失败日志记录所有可能的信息(全双工core dump)。由于他们的大小，他们在系统内存不可能存储。所以，这些日志只生成，如果指向本地设备或网络服务器日志可以存储的系统配置与URL。

失败日志是失败事件信息一个不变信息库。每个事件被编号并且包含文本关联与CPU (minicore)，网络处理器(NPU)，或者内核失败。已登录事件在/flash/crashlog2被记录到固定长度记录并且存储。

每当失败出现，此失败信息存储：

1. 事件记录在/flash/crashlog2文件(失败日志)存储。
2. 相关的minicore、NPU或者内核转储文件在/flash/crsh2目录存储。
3. 全双工core dump在用户配置的目录存储。

失败事件和Minicores的同步在管理卡之间

crashlog对其中每一张是唯一管理卡，因此，如果失败出现，当卡"8"是活跃的它将是被注册的卡"8"。后续转换不再将显示在日志的失败。为了获取此失败，交换机上一步到卡"8"必须完成。失败事件日志和转存对活动和暂挂管理卡是唯一，因此，如果失败在激活的卡出现然后失败事件日志和相关转存在仅激活的卡将存储。此失败信息不是可用的在备用卡片。每当卡切换由于在激活的卡的失败和失败信息在接管的卡不再显示，失败信息可以从当前活动卡仅获取。为了获取另一个卡的失败列表，切换再要求。为了避免此切换和从备用卡片得到失败信息，在两张管理卡之间的最新的失败信息的同步和维护要求。

到达的失败事件在待机的crashlog文件将发送对待机SMC/MIO并且保存以相似的方式。Minicore、NPU或者内核转存在激活SMC/MIO闪存需要同步到待机SMC/MMIO与rsync命令。当crashlog条目或整个列表通过CLI命令时删除，在活动和待机SMCs/MIOs应该清除。没有在内存的影响。所有失败涉及同步活动将由待机SMC/MIO卡的evlogd完成，暂挂evlogd装载，并且备用卡片有同步活动的足够的空间。所以系统的性能不会受影响。

命令

这些命令可以用于为了排除故障问题：

```
#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
```

Corefile在失败以后生成。通常操作员在外部服务器存储他们。corefile名称通常看起来象crash-
<Cardnum>-<CPU Num>-<Hex timestamp>-coree.gcrash-09-00-5593a1b8-core。

每当失败出现，此失败信息存储：

- 事件记录在/flash/crashlog2文件(失败日志)存储。
- 相关的minicore、NPU或者内核转储文件在/flash/crsh2目录存储。

摘要

所有ASR5x00软件设计处理预见的情况/事件和未预见到的情况/事件。当思科力争有完善的软件时，错误将不可避免地存在，并且失败将是可能的。所以会话恢复功能是很重要。思科的力争完美将最小化失败出现，并且会话恢复将允许会话在失败以后继续。但是，重要的是思科继续力争达到完善的软件。少量失败将降低同时发生多失败的可能性。当会话恢复无缝地愈合单个失败时，从多同时失败的恢复设计有点不同地。操作员应该很少(或从未)经历多同时失败，但是，如果这样是发生，ASR5x00设计恢复系统完整性作为最高优先级，可能在一些用户会话牺牲。