Tâches de gestionnaire de session ASR5x00 - Description de fonction, de crash, d'exécutions de reprise, et de logs de crash

Contenu

Introduction

Ce document décrit et explique la fiabilité de logiciel, la Disponibilité de service, et les caractéristiques de Basculement pour la gamme 5x00 du routeur de services d'agrégation de Cisco (ASR). Il présente la définition pour un crash de logiciel sur ASR5x00 et les effets du logiciel tombent en panne. L'article continue pour établir que même en cas de logiciel inattendu tombe en panne, comment l'ASR5x00 peut fournir le but de la Disponibilité de « classe porteuse » due à la résilience inhérente de logiciel et de la Disponibilité de caractéristiques. L'abonné mobile devrait ne jamais devoir penser à la Disponibilité du service. Le but de Cisco n'est aucune perte de session due à aucun matériel unique ou panne de logiciel, qui incluent la perte d'un système complet, en d'autres termes - fiabilité à fréquence vocale. Les caractéristiques de fiabilité de logiciel sur ASR5x00 sont visées pour pouvoir atteindre les buts pour la disponibilité des services de « classe porteuse » même dans les cas où les pannes imprévues pourraient se produire dans le réseau d'un opérateur.

Architecture logicielle : Conçu pour la résilience

L'ASR5x00 a une collection de tâches de logiciel distribuées les MILLIONS DE) cartes à travers la carte de services de paquets (PSC) ou la carte perforée (DPC) et de gestion du système carte (SMC) ou Gestion et E/S (qui sont conçues pour remplir un grand choix de fonctions spécifiques.

Par exemple, la tâche de gestionnaire de session est responsable de manipuler les sessions pour un ensemble d'abonnés et pour assurer des services intégrés tels que le peer-to-peer (P2P), Deep Packet Inspection (DPI), et ainsi de suite, sur le trafic d'utilisateur. La tâche de gestionnaire d'Authentification, autorisation et comptabilité (AAA) est responsable de la génération des

événements de facturation afin d'enregistrer l'utilisation du trafic d'abonné et ainsi de suite. Le gestionnaire de session et le gestionnaire d'AAA charge le passage sur la carte PSC/DPC.

La carte SMC/MIO est réservée pour le fonctionnement et entretien (ORGANISATION SCIENTIFIQUE DU TRAVAIL) et les tâches associées par plate-forme. Le système ASR5x00 est pratiquement compartimenté dans différents sous-systèmes de logiciel tels que le sous-système de session pour traiter des sessions d'abonné et le sous-système VPN qui est responsable de l'affectation d'adresse IP, routage, et ainsi de suite. Chaque sous-système a une tâche de contrôleur qui surveille les santés du sous-système qu'il contrôle. Le contrôleur charge le passage sur la carte SMC/MIO. Le gestionnaire de session et des tâches de gestionnaire d'AAA sont appareillés ensemble afin de manipuler la session d'un abonné pour le contrôle, le trafic de données, et la facturation. Quand la reprise de session est activée dans le système, chaque tâche de gestionnaire de session sauvegarde l'état de son ensemble d'états d'abonné avec une tâche de gestionnaire d'AAA de pair d'être récupéré en cas d'un crash de gestionnaire de session.

Quel est un crash ?

Une tâche dans l'ASR5x00 peut potentiellement tomber en panne si elle rencontre une condition de panne pendant le fonctionnement normal. Un défaut de crash ou de logiciel dans l'ASR5x00 est défini pour être une sortie ou un arrêt *inattendue d'une* tâche dans le système. Un crash peut se produire si les tentatives de code logiciel aux zones de mémoire d'accès qui sont interdites (comme les structures de données corrompues), rencontre une condition dans le code qui n'est pas prévu (comme une transition non valide d'état), et ainsi de suite. Un crash peut également être déclenché si la tâche devient insensible à la tâche de moniteur système et aux tentatives de moniteur de détruire et redémarrer la tâche. Un événement de crash peut également être explicitement déclenché (par opposition à inattendu) dans le système quand une tâche est forcée de vider son état actuel par une commande CLI ou par le moniteur système afin d'analyser l'état de tâche. Un événement prévu de crash peut également se produire quand la reprise de tâches de contrôleur système elles-mêmes potentiellement correcte une situation avec une tâche de gestionnaire qui échoue à plusieurs reprises.

Effets d'un crash de gestionnaire de session

Sous le fonctionnement normal, une tâche de gestionnaire de session manipule un ensemble de sessions d'abonné et de trafic de données associé pour les sessions avec une tâche scrutante de gestionnaire d'AAA qui manipule la facturation pour ces sessions d'abonné. Quand un crash de gestionnaire de session se produit il cesse d'exister dans le système. Si la reprise de session est activée dans le système, une tâche de réserve de gestionnaire de session est faite pour devenir active dans la même carte PSC/DPC. Cette nouvelle tâche de gestionnaire de session rétablit les sessions d'abonné pendant qu'elle communique avec la tâche de gestionnaire d'AAA de pair. L'exécution de reprise s'étend de 50 millisecondes à quelques secondes dépendantes sur le nombre de sessions qui étaient en activité dans le gestionnaire de session au moment du crash et le chargement global CPU sur la carte et ainsi de suite. Il n'y a aucune perte en sessions d'abonné qui ont été déjà établies dans le gestionnaire de session initiale dans cette exécution. N'importe quelle session d'abonné qui était en cours d'établissement au moment du crash sera vraisemblablement également due restauré aux retransmissions de protocole et ainsi de suite. Tous les paquets de données qui étaient dans la transition par le système au moment du crash peuvent assumé pour être associé avec une perte de réseau par les entités de communication de la connexion réseau et seront retransmis et la connexion seront effectués par le nouveau

gestionnaire de session. Les informations de facturation pour les sessions portées par le gestionnaire de session seront préservées dans le gestionnaire d'AAA de pair.

Quand l'opérateur devrait-il obtenir intéressé ?

Quand un crash de gestionnaire de session se produit, la procédure de récupération se produit comme décrit précédemment et le reste du système demeure inchangé par cet événement. Un crash dans un gestionnaire de session n'affecte pas les autres gestionnaires de session. Comme conseils à l'opérateur, si les tâches de gestionnaire de plusieurs sessions *sur la même carte PSC/DPC* tombent en panne simultanément ou dans un délai de 10 minutes de l'un l'autre, il pourrait y avoir perte de sessions car le système ne pourrait pas pouvoir commencer de nouveaux gestionnaires de session assez rapides pour remplacer les tâches tombées en panne. Ceci correspond à un double scénario de défaut où la perte de sessions peut se produire. Quand la reprise n'est pas faisable, le gestionnaire de session est simplement redémarré et est prêt à recevoir de nouvelles sessions.

Quand un gestionnaire de session donné tombe en panne à plusieurs reprises (comme lui rencontre la même condition de panne à plusieurs reprises), la tâche de contrôleur de session prend la note et se redémarre afin d'essayer de restaurer le sous-système. Si la tâche de contrôleur de session ne peut pas stabiliser le sous-système de session et se redémarre sans interruption plus de dans cet effort, l'étape suivante dans la transmission des problèmes est pour que le système s'oriente vers une carte du standby SMC/MIO. Dans l'événement peu probable qu'il n'y a aucune carte du standby SMC/MIO ou si une panne est produite dans l'exécution de basculement, le système se redémarre.

Les gestionnaires de session mettent à jour également des statistiques pour chaque nom de Point d'accès (APN), des services, les functionalites, et ainsi de suite qui seront de manière permanente perdus quand un crash se produit. Par conséquent une entité externe qui collecte des bulkstats périodiquement observera une immersion en statistiques quand un ou plusieurs crash se produisent. Ceci peut se manifester comme immersion dans une représentation graphique des statistiques dessinée au-dessus d'un axe de temps.

> **Note**: Un châssis typique rempli avec PSC 7-14 ou 4-10 cartes DPC a environ 120-160 gestionnaires de session, dépendants sur le nombre de cartes PSC/DPC, et un crash simple aura comme conséquence la perte d'environ $1/40^{Th}$ ou de $1/80^{Th\ des}$ statistiques. Quand un gestionnaire de session de réserve succède, il commence à accumuler les statistiques de nouveau de zéro.

Comment savoir si un crash se produisait ?

Un crash déclenchera un événement de déroutement SNMP à une station de Surveillance de réseau, telle que le service de contrôle d'événement (SME) et par des événements de Syslog. On peut également observer les crash qui se sont produits dans le système avec la commande de **liste de crash d'exposition**. Notez que des listes de ces commandes inattendues et des événements prévus de crash comme décrit plus tôt. Ces deux types d'événements de crash peuvent être distingués au moyen d'une en-tête qui décrit chaque crash.

Un crash de tâche suivi de reprise réussie de session est indiqué par ce message de log :

```
"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#>"
```

Un crash de tâche qui ne pourrait pas récupérer est indiqué par ce message de log :

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

En résumé, avec la reprise de session activée, dans la plupart des cas les crash ne seront pas notés parce qu'ils n'ont aucune incidence d'abonné. On doit sélectionner la commande CLI, ou regardez les logs ou la notification SNMP afin de détecter n'importe quelle occurrence des crash.

Exemple :

```
 ******** 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 se connectant l'architecture

Les logs de crash enregistrent toutes les informations possibles qui concernent un crash de logiciel (plein vidage de mémoire). En raison de leur taille, ils ne peuvent pas être enregistrés dans la mémoire système. Par conséquent, ces logs sont seulement générés si le système est configuré avec un URL qui indique un périphérique local ou un serveur de réseau où le log peut être enregistré.

Le log de crash est un référentiel persistant des informations sur l'événement de crash. Chaque événement est numéroté et contient le texte associé avec une CPU (minicore), l'unité de traitement réseau (NPU), ou le crash de noyau. Les événements loggés sont enregistrés dans des enregistrements de longueur fixe et enregistrés dans /flash/crashlog2.

Toutes les fois qu'un crash se produit, ces informations de crash sont stockées :

1. L'enregistrement d'événement est enregistré dans le fichier de /flash/crashlog2 (le log de crash).
2. Le minicore, le NPU, ou le fichier associé de vidage mémoire de noyau est enregistré dans le répertoire de /flash/crsh2.
3. Un plein vidage de mémoire est enregistré dans un répertoire configuré par utilisateur.

Synchronisation des événements et de Minicores de crash entre les cartes de gestion

Le crashlog est seul à chacune des cartes de gestion, ainsi si un crash se produit quand la carte "8" est en activité ce sera la carte ouverte une session "8". Un basculement ultérieur n'afficherait plus le crash dans le log. Afin de récupérer ce crash, un commutateur de retour plus de pour carder "8" doit être fait. Le journal d'événements et les vidages mémoire de crash sont seuls aux cartes de gestion actives et de réserve, ainsi si un crash se produit sur une carte à puce puis le journal d'événements de crash et les vidages mémoire relatifs seront enregistrés sur une carte à puce seulement. Ces informations de crash ne sont pas disponibles sur la carte de réserve. Toutes les fois que le basculement de cartes dû à un crash dans la carte à puce, et les informations de crash n'est plus affiché sur la carte qui succède, les informations de crash peuvent être récupérées seulement de la carte à puce en cours. Afin de récupérer la liste de crash de l'autre carte, un basculement est exigé de nouveau. Afin d'éviter ce basculement et obtenir les informations de crash de la carte de réserve, la synchronisation entre deux cartes de gestion et la maintenance des dernières informations de crash est exigée.

L'événement de arrivée de crash sera envoyé plus d'au standby SMC/MIO et enregistré dans le fichier du crashlog du standby de la manière semblable. Minicore, NPU, ou vidages mémoire de noyau sur l'éclair de SMC/MIO actif doit être synchronisé au standby SMC/MMIO avec la commande de **rsync**. Quand une entrée de crashlog ou la liste de totalité est supprimée par la commande CLI, elle devrait être effacée en actif et état d'alerte SMC/MIOs. Il n'y a aucune incidence sur la mémoire. Toute l'activité relative par crash de synchronisation sera faite par l'evlogd de la carte du standby SMC/MIO, car l'evlogd de réserve moins est chargé et la carte de réserve a assez de pièce pour l'activité de synchronisation. Par conséquent les performances du système ne seront pas affectées.

Commandes

Ces commandes peuvent être utilisées afin de dépanner des questions :

```
#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 sont générés après un crash. Habituellement les opérateurs les enregistrent dans un serveur externe. Le nom corefile ressemble à habituellement le crash-<Cardnum>-<CPU Num>-<Hex timestamp>-coree.gcrash-09-00-5593a1b8-core.

Toutes les fois qu'un crash se produit, ces informations de crash sont stockées :

- L'enregistrement d'événement est enregistré dans le fichier de /flash/crashlog2 (le log de crash).
- Le minicore, le NPU, ou le fichier associé de vidage mémoire de noyau est enregistré dans le répertoire de /flash/crsh2.

Résumé

Tout les logiciel ASR5x00 est conçu pour manipuler des conditions prévues/événements et des conditions/événements imprévus. Tandis que Cisco tâche d'avoir le logiciel parfait, inévitablement les erreurs existeront et les crash seront possibles. C'est pourquoi la caractéristique de reprise de session est si importante. Cisco essayent d'obtenir la perfection réduira les occurrences des crash, et la reprise de session permettra aux sessions pour continuer après un crash. Néanmoins, il est important que Cisco continue à tâcher de réaliser le logiciel parfait. Moins crash réduiront la probabilité des plusieurs crash qui se produisent simultanément. Tandis que la reprise de session guérit sans faille un crash simple, la reprise de plusieurs crash simultanés est conçue un bit différemment. Les opérateurs devraient rarement (ou) ne jamais éprouver de plusieurs crash simultanés, mais si tels étaient de se produire, l'ASR5x00 est conçu pour récupérer l'intégrité de système en tant que plus prioritaire, probablement au sacrifice de quelques sessions d'abonné.