Comment identifier un redémarrage Cisco CallManager comme panne ou arrêt de service

Contenu

Introduction

Ce document décrit la différence entre un crash de Cisco CallManager et un arrêt de service. Le document fournit également la procédure pour signaler un crash de Cisco CallManager et pour permettre au support technique de Cisco de dépanner la question.

Conditions préalables

Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de logiciel suivantes :

- Cisco CallManager 3.x et 4.0

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à Conventions relatives aux conseils techniques Cisco.

Différence entre les crash de Cisco CallManager et les arrêts

Crash

Une bogue dans le code de Cisco CallManager entraîne des crash de CallManager. Il y a trois types principaux de crash :

- Violations d'Access
- Clivage par zéro
- Exceptions inconnues

Les crash génèrent les entrées de Dr. Watson, qui sont ajoutées à l'extrémité du fichier existant de Dr. Watson. Les crash génèrent également des fichiers user.dmp. L'emplacement de ces fichiers est des utilisateurs de C:\Documents and Settings\All \ documents \ DrWatson.

Le nom du fichier de Dr. Watson, qui est un fichier texte, est drwtsn32.log.

Choisissez **drwtsn32 de la** fenêtre de passage pour configurer les configurations.

Comment lire un Dr. Watson File

Terminez-vous ces étapes pour lire le fichier de Dr. Watson :

1. Recherchez le mot « quand », en minuscules, et trouvez la date et l'heure auxoù le problème s'est posé.Le fichier de Dr. Watson enregistre tous les crash d'application. Quelques articles blocage système peuvent ne pas être des crash de Cisco CallManager. Les exemples des articles blocage système qui ne sont pas des crash de Cisco CallManager incluent RisDC.exe et aupair.exe.
2. Après que vous localisiez la date et l'heure du problème, localisez le nombre de processus de l'identifiant (PID), et recherchez la liste des tâches pour déterminer quelle application est tombée en panne.La liste des tâches apparaît dans l'exemple dans cette étape.Dans cet exemple, l'application qui est tombée en panne a un PID de 752, et le nom de l'application est SCAN32.exe :
```
Application exception occurred:
        App:  (pid=752)         When: 9/1/2000 @ 10:23:40.836         Exception number:
c0000005 (access violation) *----> System Information <----*         Computer Name: CISCO-
8VCUWBLUJ         User Name: SYSTEM         Number of Processors: 1         Processor Type:
x86 Family 6 Model 8 Stepping 3         Windows 2000 Version: 5.0         Current Build:
2195         Service Pack: None         Current Type: Uniprocessor Free         Registered
Organization: Cisco Systems Inc.         Registered Owner: Cisco User *----> Task List <---
-*    0 Idle.exe    8 System.exe  144 smss.exe  168 csrss.exe  164 winlogon.exe  216
services.exe  228 lsass.exe  336 ibmpmsvc.exe  380 svchost.exe  424 svchost.exe  576
regsvc.exe  592 MSTask.exe  924 Explorer.exe  992 cmd.exe  972 msiexec.exe  928 tp4mon.exe
 856 ibmpmsvc.exe  852 ltmsg.exe  408 RunDll32.exe  428 RunDll32.exe  328 PDirect.exe  620
TP98.exe  968 tphkmgr.exe  948 PRPCUI.exe  668 AUTOCHK.exe  744 tponscr.exe  868 KIX32.exe
 520 spoolsv.exe 1164 Avsynmgr.exe 1136 VsStat.exe 1192 Vshwin32.exe 1224 Mcshield.exe 1024
MCUPDATE.exe 752 SCAN32.exe 1292 drwtsn32.exe    0 _Total.exe
```
3. Si le crash est un crash de Cisco CallManager, notez le nombre d'exception pour déterminer le type de crash.**Remarque:** Artère à l'équipe de développement appropriée un crash d'application qui n'est pas un crash de Cisco CallManager, s'il y a lieu.
```
Application exception occurred:

        App:  (pid=752)
```

When: 9/1/2000 @ 10:23:40.836

Exception number: **c0000005 (access violation)** Dans cet exemple, le nombre d'exception est c0000005, qui est une violation d'accès. Cette violation d'accès signifie que l'application a tenté à la mémoire d'accès en dehors de la limite de mémoire d'application qui le positionnement de système d'exploitation.Un autre type possible de crash est clivage par zéro. Le comme indiqué dans cet exemple, le nombre d'exception pour le clivage par zéro est c0000094 :Application exception occurred:

App: (pid=1564)

When: 1/7/2003 @ 13:16:15.051

Exception number: **c0000094 (divide by zero)** Le type de crash peut également être exception inconnue. Le comme indiqué dans cet exemple, le nombre d'exception pour l'exception inconnue est e06d7363 :Application exception occurred:

App: (pid=2784)

When: 12/10/2002 @ 09:02:58.202

Exception number: **e06d7363** Quand vous déterminez si un crash est une violation d'accès, clivage par zéro, ou exception inconnue, vous pouvez apparier un crash avec une bogue Cisco existante. Si vous ne trouvez aucune correspondance, l'ingénieur de développement a un bon début pour déterminer ce qui s'est produit.

4. Recherche sous quand section du fichier pour le mot DÉFAUT à commencer à définir la « signature » du crash. **Remarque:** Le DÉFAUT apparaît en majuscules.Cette section de DÉFAUT du fichier contient six informations les informations importantes, qui sont :Le nombre du thread qui a rencontré le problèmeLe contenu des inscriptions à ce thread au moment du crashLa fonction qui a exécuté au moment du crashLa déclaration de code d'assemblage qui a eu comme conséquence le crashLa pile suivent qui affiche les adresses des fonctions qu'exécutée, dans la commande, juste avant le crashLe vidage mémoire cru de pile, qui fournit plus d'informations au sujet de ce qui était sur la pile d'exécution au moment du crashCe code fournit un exemple d'un crash de Cisco CallManager qui est un crash de violation d'accès. Le texte en caractères gras met en valeur les six éléments indispensables, aussi bien que le mot DÉFAUT, qui marque cette section du fichier :State Dump for **Thread Id 0x998** *!--- This number is the number of the thread that experienced the problem.*
eax=00cae82c ebx=02070000 ecx=00e95da0 edx=346984d8 esi=34698970 edi=346984d8 eip=77fcb9b3
esp=05cef34c ebp=05cef358 iopl=0 nv up ei ng nz na pe cy cs=001b ss=0023 ds=0023
es=0023 fs=003b gs=0000 efl=00000283 *!--- This provides the contents of the registers at the time of the crash.* **function: RtlSizeHeap** *!--- This function executed at the time of the crash.* 77fcb992 0f87aafeffff jnbe RtlFreeHeap+0x20f
(77fcb842) 77fcb998 807d1400 cmp byte ptr [ebp+0x14],0x0 ss:
0650c92a=?? 77fcb99c 0f8586300000 jne RtlZeroHeap+0x327 (77fcea28)
77fcb9a2 57 push edi 77fcb9a3 53 push ebx
77fcb9a4 e8646cfbff call RtlConsoleMultiByteToUnicodeN+0x348 (77f8260d)
77fcb9a9 8b4f0c mov ecx,[edi+0xc] ds: 34eb5aaa=00003781
77fcb9ac 8b4708 mov eax,[edi+0x8] ds: 34eb5aaa=00003781
77fcb9af 3bc1 cmp eax,ecx 77fcb9b1 8901 mov
[ecx],eax ds: 00e95da0=00cae82c **FAULT ->77fcb9b3 894804 mov
[eax+0x4],ecx ds: 014cbdfe=ec810000** *!--- This is the assembly code statement that resulted in the crash.* 77fcb9b6 744d jz 77fd4405 77fcb9b8
8a4705 mov al,[edi+0x5] ds: 34eb5aaa=81 77fcb9bb
a804 test al,0x4 77fcb9bd 0f8521310000 jne RtlZeroHeap+0x3e3
(77fceae4) 77fcb9c3 8a4605 mov al,[esi+0x5] ds:
34eb5f42=d5 77fcb9c6 2410 and al,0x10 77fcb9c8

```
a810             test    al,0x10         77fcb9ca 884705           mov
[edi+0x5],al                ds: 34eb5aaa=81         77fcb9cd 0f8555030000     jne
RtlSizeHeap+0x3ef (77fcbd28)         77fcb9d3 0fb70f           movzx   ecx,word ptr
[edi]         ds: 346984d8=0093         77fcb9d6 8b4510           mov
eax,[ebp+0x10]         ss: 0650c92a=???????? *----> Stack Back Trace <----* !--- This
shows, in order, the addresses of the functions that executed !--- just before the crash.
FramePtr ReturnAd Param#1  Param#2  Param#3  Param#4  Function Name 05CEF358 77FCB733
02070000 34698970 05CEF3D0 00000000 ntdll!RtlSizeHeap 05CEF400 7800115C 02070000 00000000
34698978 05CEF454 ntdll!RtlFreeHeap 05CEF448 00C0304F 34698978 00545EC2 34698978 34698978
!free 05CEF460 00B66F85 00000001 00B6626C 033B3D58 025A6720 !<nosymbols> 05CEFF34 018E736B
025A6720 77E964CB 033C6B20 033C6B20 !<nosymbols> 05CEFF80 780060CE 033B3D58 77E964CB
00000018 033C6B20 !ACE_OS_Thread_Adapter:: invoke µµ 05CEFFEC 00000000 00000000 00000000
00000000 00000000 kernel32!TlsSetValue *----> Raw Stack Dump <----* !--- This provides more
information about what was on the run-time stack !--- at the time of the crash. 05cef34c
00 00 07 02 70 89 69 34 - 00 00 00 00 00 f4 ce 05  ....p.i4........ 05cef35c  33 b7 fc 77
00 00 07 02 - 70 89 69 34 d0 f3 ce 05  3..w....p.i4.... 05cef36c  00 00 00 00 54 f4 ce 05 -
78 89 69 34 20 67 5a 02  ....T...x.i4 gZ. 05cef37c  44 5b e3 09 94 f3 ce 05 - 30 e6 b5 00
fc f3 ce 05  D[......0....... 05cef38c  38 29 6a 09 40 5b e3 09 - a8 f3 ce 05 65 e5 b5 00
8)j.@[......e... 05cef39c  fc f3 ce 05 38 29 6a 09 - 40 5b e3 09 c4 f3 ce 05
....8)j.@[...... 05cef3ac  39 e2 b5 00 57 92 89 01 - 30 db 55 02 f5 50 5b 00
9...W...0.U..P[. 05cef3bc  e0 f3 ce 05 cc f3 ce 05 - 0f 4f 5b 00 e0 f3 ce 05
.........O[..... 05cef3cc  00 00 07 02 19 00 00 00 - 01 f4 ce 05 f8 2b cf 21
.............+.! 05cef3dc  f8 2b cf 21 01 f1 ce 05 - 28 ff ce 05 70 f3 ce 05
.+.!....(...p... 05cef3ec  98 ef ce 05 38 f4 ce 05 - a7 9d fb 77 90 26 f8 77
....8......w.&.w 05cef3fc  01 00 00 00 48 f4 ce 05 - 5c 11 00 78 00 00 07 02
....H...\..x.... 05cef40c  00 00 00 00 78 89 69 34 - 54 f4 ce 05 04 fa ce 05
....x.i4T....... 05cef41c  20 67 5a 02 02 00 00 00 - 64 00 00 00 5c 00 00 00
gZ.....d...\... 05cef42c  fe 08 00 00 00 00 00 00 - 98 ef ce 05 28 ff ce 05
............(... 05cef43c  b8 ff 00 78 50 32 03 78 - ff ff ff ff 60 f4 ce 05
...xP2.x....`... 05cef44c  4f 30 c0 00 78 89 69 34 - c2 5e 54 00 78 89 69 34
O0..x.i4.^T.x.i4 05cef45c  78 89 69 34 34 ff ce 05 - 85 6f b6 00 01 00 00 00
x.i44....o...... 05cef46c  6c 62 b6 00 58 3d 3b 03 - 20 67 5a 02 98 f6 e6 36  lb..X=;.
gZ....6 05cef47c  98 f6 e6 36 00 00 00 00 - 00 00 00 00 00 00 00 00  ...6............
```

Ces six bits des informations constituent une partie de, mais pas toutes de, la signature de crash. Le reste des informations qui définissent la signature est :Le type de crash (violation d'accès, clivage par zéro, ou exception inconnue)Les dernières déclarations de suivi de la couche de distribution de signal (SDL) qui ont exécuté avant le crash**Remarque:** Le dernier fichier SDL qui a eu l'utilisation avant le crash, aussi bien que le fichier de Dr. Watson, des attachés à n'importe quel crash introduisent des erreurs pour tests.Les attachés de la cette information de signature (le dernier fichier SDL, le dernier fichier de Cisco CallManager, et le fichier de Dr. Watson) au système de recherche réparti de défaut (DDTS) enregistrent quand vous créez un nouveau crash DDTS.Si vous appariez le nouveau crash avec un DDTS qui déjà existe et a la même cause principale, ces informations sont identique :Le type d'exceptionLe nom de la fonction qui a exécuté au moment du crashLes noms des fonctions dans la pile suivent**Remarque:** Ces noms n'apparaissent pas toujours dans un fichier de Dr. Watson.La déclaration de code d'assemblage qui apparaît à côté du repère de DÉFAUTLes dernières lignes de suivi SDL, qui devraient être très semblablesLe contenu des registres, des adresses mémoire, et d'autres informations peut différer des informations dans un autre DDTS qui existe, même si le crash a la même cause principale. Les adresses varient si vous exécutez une différente version de Cisco CallManager. Si vous exécutez la même version du Cisco CallManager, les adresses dans la section de code d'assemblage et dans la section de suivi de pile sont identiques.

5. Collectez ces fichiers pour mettre au point le crash :drwtsn32.loguser.dmpLes derniers fichiers de suivi SDL et de Cisco CallManager, d'approximativement 5 minutes avant le crash et de 5 minutes après la reprise.fichiers de proglog**Remarque:** Collectez ces fichiers seulement dans des versions 3.2 et ultérieures de Cisco CallManager.Journaux

d'événements, système et application, si disponible.Logs de moniteur de performances (perfmon), si disponible.

Erreur de DBLException

Vous voyez ce message d'erreur dans le journal d'application du Cisco CallManager Publisher et de l'abonné :

```
Error: DBLException - DBL Exception.
  ErrorCode: 8
  ExceptionString: Invalid parameter
  UNKNOWN_PARAMNAME:Text: addDevice
  App ID: Cisco CallManager
  Cluster ID: XXXX-Cluster
  Node ID: 192.168.0.2
Explanation: Severe database layer interface error occurred.
Recommended Action: Contact TAC..
```

Ou :

```
A COM error occurred during processing. (6)

Details:
Error No. -2147219962 (0x80040606):
CDBLException Dump: [COM Error] COM Error Description = []
```

Ce type d'erreur se produit quand un téléphone IP est rejeté de l'enregistrement ou en raison d'un abonnement cassé entre l'éditeur et les bases de données d'abonné. Ce problème peut être résolu à l'aide de l'outil de DBLHelper. Pour plus d'informations sur DBLHelper, référez-vous en employant DBLHelper pour rétablir un abonnement cassé de la batterie SQL de Cisco CallManager.

Cette erreur peut également se produire en raison du crash de service du moniteur de couche de Cisco Database. Exécutez ces étapes pour résoudre le problème :

1. Allez au **Start > Programs > Administrative tools** > des **services composants**.
2. Développez les **services** > les **ordinateurs composants > mon ordinateur > applications COM+**.
3. Commencez le service **MSDTC** (coordonnateur distribué de transaction), s'il affiche arrêté.

Arrêts

L'autre type de reprise de Cisco CallManager est un arrêt. Un arrêt est quand le Cisco CallManager ne peut pas utiliser efficacement et se ferme. Les arrêts se rangent dans deux catégories :

- Délais d'attente d'initialisation
- Temporisateur SDL et fataux de thread de routeur SDL

Si le Cisco CallManager se ferme, vous trouvez `code de raison d'`arrêt dans les dernières lignes de suivi du suivi de CallManager. Voici un exemple :

```
03/22/2003 14:32:16.562 Cisco CallManager|CallManagerFailure - Indicates
some failure in the Cisco CallManager system. Host name of hosting
node.:NEROCM1 IP address of hosting node.:172.27.27.224 Reason code.:4 Additional Text
[Optional]: App ID:Cisco CallManager Cluster ID:NEROCM1-Cluster Node
ID:172.27.27.224|<CLID::NEROCM1-Cluster><NID::172.27.27.224><CT::Alarm>
```

Dans cet exemple, code de raison est `4`. Cette liste fournit codes de raison d'arrêt du code de Cisco CallManager :

```
class CallManagerFailureAlarm : public CallManagerAlarmCatalog {
public:
            enum Reason {
                        Unknown = 1,
                        HeartBeatStopped = 2,
                        RouterThreadDied =3 ,
                        TimerThreadDied = 4,
                        CriticalThreadDied = 5,
                        DeviceMgrInitFailed = 6,
                        DigitAnalysisInitFailed = 7,
                        CallControlInitFailed = 8,
                        LinkMgrInitFailed = 9,
                        DbMgrInitFailed = 10,
                        MsgTranslationInitFailed = 11,
                        SupServiceInitFailed = 12,
                        DirectoryInitFailed = 13
            };
```

La raison 1 et la raison 2 sont des rares cas d'arrêts internes, alors que les autres raisons sont plus communes. La raison 3 indique que le thread de routeur SDL a arrêté la réponse. La raison 4 indique que le thread de temporisateur SDL a arrêté la réponse. Les raisons 5 – 13 associent au feu de temporisateur d'initialisation.

Délais d'attente d'initialisation

Quand de Cisco CallManager de service les débuts d'abord, les débuts de thread de moniteur de processus de CallManager (CMProcMon). Puis, les débuts de thread de MmmanInit, qui engendre tous les autres processus. Ensuite, les débuts de thread de routeur SDL. Ce thread manipule les signaux qui envoient d'un processus à l'autre. Chacun des trois du début de thread en même temps. Le thread de CMProcMon et le thread de routeur SDL sont en hausse et fonctionnent tandis que le thread de MmmanInit commence d'autres processus.

CMProcmon et SDL doivent être en service tandis que MmmanInit commence de divers processus. Le thread de MmmanInit commence ces processus, dans cette commande :

1. Base de données (ProcessDb)**Remarque:** ProcessDb est une interface de Cisco CallManager au code de couche de base de données (DBL).En même temps, le code de MmmanInit commence également un certain nombre d'autre Cisco CallManager interne, des processus indépendants. Ces processus incluent H225Handler, MGCPBhHandler, et chef hiérarchique.
2. Régions
3. AARNeighborhood
4. Emplacements
5. Plan de routage
6. Analyse de chiffre
7. Contrôle d'appel
8. Services supplémentairesLes caractéristiques incluent le parc d'appel, en avant, la conférence, et le transfert.
9. Périphérique
10. Répertoire
11. Appelant le gestionnaire d'espace de recherche (CSSManager)

12. Gestionnaire d'heure (TODManager)

La réalisation de ces tâches se produit dans une gamme. Chacune des douze tâches a un temporisateur qui s'associe avec la tâche. Débuts de ce temporisateur quand la tâche commence. Si les feux de temporisateur avant que la tâche se termine, Cisco CallManager arrête et imprime un suivi SDL qui lit :

```
Critical thread death: name of the timer which fired
```

Cette liste affiche chacun des temporisateurs, aussi bien que le signal SDL qui commence le temporisateur et le signal SDL qui arrête le temporisateur. Les signaux de « InitDone » apparaissent dans le suivi SDL si vous avez des niveaux de set trace convenablement. (Vous avez placé SdlTraceTypeFlags à 0x8000CB15.)

Ces minuteurs par défaut sont basés sur la version 4.1(2) de Cisco CallManager. Si vous exécutez la différente version, pourrait être légèrement différent.

1. Le temps d'initialisation de base de données (par défaut à 900 secondes) - le signal de départ pendant ce temps est le signal de « début » envoyé au processus de MmmanInit. Vous voyez ceci dans le suivi SDL.
2. Temps d'initialisation de régions (par défaut à 120 secondes).
3. Temps d'initialisation d'AARNeighborhoods (par défaut à 90 secondes).
4. Temps d'initialisation d'emplacements (par défaut à 90 secondes).
5. Temps d'initialisation de plan de routage (par défaut à 600 secondes).
6. Temps d'initialisation d'analyse de chiffre (par défaut à 900 secondes).
7. Temps d'initialisation de Contrôle d'appel (par défaut à 90 secondes).
8. Le temps d'initialisation supplémentaire de services (par défaut à 900 secondes) - le signal de départ est CcInitDone et le signal d'extrémité est SsInitDone.
9. Temps d'initialisation de périphérique (par défaut à 360 secondes).
10. Temps d'initialisation de répertoire (par défaut à 90 secondes)
11. Temps d'initialisation de CSSManager (par défaut à 900 secondes).
12. Temps d'initialisation de TODManager – (par défaut à 900 secondes).

Après tout les tâches sont complètes, Cisco CallManager ouvre des liens SDL aux services de CallManager ce passage sur d'autres Noeuds dans le réseau. Le Cisco CallManager ouvre également des liens SDL aux services de gestionnaire du couplage de la téléphonie et de l'informatique (CTI) qu'exécuté sur le mêmes noeud ou différents Noeuds dans le réseau.

Puis, MmmanInit envoie un signal de CMInitComplete de nouveau au thread de CMProcMon. Quand de CMProcMon les débuts d'abord, il met en marche un temporisateur dur-codé par 60-minute pour l'initialisation de Cisco CallManager. Le temporisateur a le nom CMInitComplete_WaitTime. (Ce temporisateur n'est pas un paramètre de service ; le temporisateur n'est pas configurable.) Si le thread de CMProcMon ne reçoit pas le signal de CMInitComplete dans un délai de 60 minutes, le Cisco CallManager arrête et émet une déclaration de suivi qui lit :

```
Timed out waiting for CMInitComplete signal
```

Si des n'importe quelles des douze tâches d'initialisation échouent, ou si le temps total pour ces tâches dépasse 60 minutes, le Cisco CallManager arrête.

Remarque: Le temporisateur de CMInitComplete_WaitTime était par le passé dur codé à 10 minutes. Ce code dur a changé à 60 minutes en tant qu'élément de l'ID de bogue Cisco CSCdx31622 (clients enregistrés seulement). La modification a écrit le 3.1(3) machinant la série (es) spéciale, avec es 38 comme début. La modification est également dans 3.2(2) la série es,

avec es 11 comme début, et dans le Cisco CallManager 3.3.

Si vous éprouvez les problèmes avec un temporisateur d'initialisation se déclenchent, vous pourraient seulement devoir augmenter le paramètre du temporisateur pour résoudre le startup. Si cette modification ne résout pas le problème, le problème peut être un temps de réponse lent de base de données qui fait chronométrer l'exécution. Collect a détaillé des suivis de DBL, aussi bien que des suivis SDL et de Cisco CallManager, s'il y a lieu.

Collectez ces fichiers pour mettre au point un problème d'initialisation :

- Suivi détaillé de Cisco CallManager
- Suivi SDL**Remarque:** Placez le SdlTraceTypeFlags à 0x8000CB15.
- Suivi détaillé de DBL

Temporisateur SDL et fataux de thread de routeur SDL

Le thread de routeur SDL est le thread le plus important de l'exécution à l'intérieur de l'application de Cisco CallManager. Il contrôle l'envoi des signaux de Traitement des appels. Le thread de CMProcMon vérifie les santés du thread de routeur SDL une fois toutes les deux secondes. Les suivis de Cisco CallManager affichent cette vérification de l'intégrité, comme vous pouvez voir dans ces déclarations :

```
02/05/2003 00:30:32.790 Cisco CallManager|CMProcMon - ------Entered Router
Verification|<CLID::USNYTSVOIPPB01-Cluster><NID::10.2.40.11>

02/05/2003 00:30:32.790 Cisco CallManager|CMProcMon - ----Exited Router
Verification|<CLID::USNYTSVOIPPB01-Cluster><NID::10.2.40.11>
```

Si le thread de CMProcMon écrit et quitte la vérification de routeur, le thread de routeur SDL répondu à la vérification de l'intégrité et est bien.

Cependant, si le thread de routeur SDL ne répond pas, vous voyez `tandis que boucle` dans le suivi de Cisco CallManager, en tant que ce des expositions :

```
CMProcMon - ----Entered While loop ++++ TimeAtWhileEntry: [some number here],
TimeBeforeSleep: [another number], TimeAfterSleep: [a third number], sleepTimeWas :
[4th number"
```

Dans cette situation d'urgence, le thread de routeur SDL reçoit des contrôles une fois chaque seconde pendant une période de 20 secondes. Si le thread répond à tout moment au cours de la période 20-second, le fonctionnement normal reprend, et la santé du thread de routeur SDL reçoit de nouveau la vérification toutes les 2 secondes. Si, cependant, le thread de routeur SDL ne répond pas aux contrôles au cours des 20 secondes, l'application de Cisco CallManager s'est arrêtée. Cette déclaration apparaît dans le suivi SDL :

```
000177508| 01/12/31 07:28:40.389| 001| AlarmErr  |
 |                  |                  |                  |
| AlarmClass: CallManager, AlarmName: CallManagerFailure, AlarmSeverity: Error
AlarmMessage: , AlarmDescription: Indicates some failure in the Cisco CallManager
 system.,
AlarmParameters:  HostName:CCM-PUB, IPAddress:10.5.162.180, Reason:3, Text:, AppID:Cisco
CallManager, ClusterID:CCM-PUB-Cluster, NodeID:10.5.162.180,
```

Notez code de raison `3` dans le texte de cette déclaration de suivi. Le code signifie que le thread de routeur SDL a arrêté la réponse, ainsi le Cisco CallManager s'est arrêté.

La cause le plus susceptible d'un arrêt de thread de routeur SDL est un manque de ressources

système. Une autre application a utilisé les la plupart ou toute les CPU pendant une longue période de temps, au moins 20 secondes. Cette activité est pourquoi les moniteurs de performances sont essentiels de mettre au point ce type d'arrêt.

L'autre type d'arrêt à l'explorer est l'arrêt de thread de temporisateur SDL. Un arrêt de thread de temporisateur SDL se produit quand le différentiel entre l'horloge interne de Cisco CallManager et l'horloge du système d'exploitation externe dépasse 16 secondes. Quand l'arrêt de thread de temporisateur SDL se produit, ce suivi apparaît dans le suivi de Cisco CallManager :

```
03/22/2003 14:32:16.562 Cisco CallManager|CallManagerFailure - Indicates
some failure in the Cisco CallManager system. Host name of hosting
node.:NEROCM1 IP address of hosting node.:172.27.27.224 Reason code.:4 Additional Text
[Optional]: App ID:Cisco CallManager Cluster ID:NEROCM1-Cluster Node
ID:172.27.27.224|<CLID::NEROCM1-Cluster><NID::172.27.27.224><CT::Alarm>
```

Le Cisco CallManager vérifie généralement les thread de temporisateur une fois chaque seconde. Le Cisco CallManager ajoute 1 seconde au temps du système d'exploitation en cours et les mémoires qui évaluent loin en tant que « temps prévu. » Puis, le Cisco CallManager dort pour 1 seconde. Après que le Cisco CallManager se réveille, il vérifie le nouveau temps du système d'exploitation et soustrait le temps prévu. Si la différence entre ces deux moments est plus de 1 seconde, cette déclaration d'avertissement apparaît dans le suivi de Cisco CallManager :

```
CMProcMon::star_sdlVerification - Test Timer exceeded minimum timer latency
threshold of  1000 milliseconds, Actual latency: 1630 milliseconds
```

`La latence réelle` dans cette déclaration prouve que le thread interne de temporisateur du Cisco CallManager SDL exécute lent. Ici, la différence entre le Cisco CallManager a attendu le temps et le temps du système d'exploitation réel est de 1.63 seconde.

Si cette différence dépasse 16 secondes, le Cisco CallManager a arrêté et fournit code de raison d'arrêt de `4`.

La cause le plus susceptible d'un arrêt de thread de temporisateur SDL est un manque de temps-CPU pour le Cisco CallManager. Une autre application, telle que VirusScan ou sauvegarde STI, a utilisé la plupart des ressources CPU pendant au moins 16 secondes. Les logs de Perfmon sont essentiels de déterminer la cause principale de ce type d'arrêt.

Si la sauvegarde d'applications de Téléphonie sur IP de Cisco fonctionne pendant une longue période de temps à l'utilisation du CPU élevé, un blocage système peut se produire. Pour les informations sur la façon dont éviter ce blocage système, référez-vous :

- Configurations de contrôle sur l'utilitaire de sauvegarde pour éviter la section CPU de haute du crash de service de Cisco CallManager de document

Collectez ces fichiers dans le cas d'un arrêt de thread de thread de routeur SDL ou de temporisateur SDL :

- Suivi détaillé de Cisco CallManager
- Suivi SDL**Remarque:** Placez le SdlTraceTypeFlags à 0x8000CB15.
- Perfmon trace, si disponible, qui affichent l'utilisation du processeur de pour cent de tous les processus qui fonctionnent sur la case**Remarque:** Vous pouvez capturer ces suivis à distance pour réduire l'incidence des performances sur la CPU du serveur Cisco CallManager.

Comment signaler des crash de Cisco CallManager au support technique de Cisco

Le diagnostic de la cause réelle d'un crash de Cisco CallManager est difficile. Afin de déterminer la cause et accélérer une solution, le support technique de Cisco exige de vous de collecter des suivis et des logs de Dr. Watson et de télécharger les informations aux notes en cas de support technique de Cisco. Vous envoyez les notes en cas à attach@cisco.com et fournissez le numéro de dossier dans le champ objet de courrier électronique. La procédure est :

1. Collectez les fichiers de suivi de Cisco CallManager de 30 minutes avant et de 15 minutes après le crash.L'emplacement des suivis est C:\Program Files\Cisco\Trace\CCM.
2. Collectez les fichiers de suivi SDL de 30 minutes avant et de 15 minutes après le crash.L'emplacement des suivis est C:\Program Files\Cisco\Trace\SDL\CCM.
3. Collectez les fichiers user.dmp et de drwtsn32.log.L'emplacement des fichiers est des utilisateurs de C:\Documents and Settings\All \ documents \ DrWatson.
4. **Start > Programs > Administrative tools > visualisateur d'événements** choisis fichiers de consignation pour recueillir de système et d'application journal d'événements du visualisateur d'événements.Vous pouvez ignorer cette étape, si les données de journal d'événements ne sont pas nécessaires. Mais, videz les événements de système et d'application et filtrez seulement les événements d'approximativement 30 minutes avant le crash. Étudiez ces événements avant que vous les envoyiez au support technique de Cisco. Vous pouvez trouver un événement qui justifie plus d'attention.**Attention :** Faites attention si vous utilisez le visualisateur d'événements, un utilitaire intégré de Microsoft, à vider ces événements à un fichier texte. Dans un système qui a l'utilisation du CPU élevé, cette utilisation de visualisateur d'événements peut facilement mourir de faim tous autres processus de la CPU. Ces processus incluent le processus de keepalive de Cisco CallManager qui met à jour des enregistrements de téléphone.Vous pouvez employer un utilitaire de shareware avec le nom elogdmp.exe pour vider toutes les entrées dans les différents logs à un fichier texte. Les implications CPU sont négligeables quand vous utilisez l'outil elogdmp.exe. Émettez cette commande d'une invite DOS :
   ```
   elogdmp COMPUTER_NAME Application > AppEvents.txt elogdmp COMPUTER_NAME System > SysEvents.txt
   ```
5. Compressez tous les fichiers comme fichier zip dans la commande que cette étape affiche avant que vous courrier électronique et copie les fichiers.Version 8 de WinZip d'utilisation pour compresser les fichiers. (Cisco a une licence de site pour cet utilitaire.) Généralement copie de fichiers à un ordinateur local pour une évaluation plus rapide. Les fichiers que vous compressez l'utilisation moins d'espace, et vous pouvez déplacer ces fichiers beaucoup plus rapidement que des formats de fichier brut.Compressez les fichiers user.dmp et de drwtsn32.log ensemble.Immédiatement envoyez et copiez ce fichier zip. Fournissez une définition descriptive de symptôme et incluez la version de Cisco CallManager, les chargements de périphérique approprié, et les versions de logiciel précis de Cisco IOS®. Si des correctifs spéciaux sont en service, assurez-vous que vous rendez ce fait clair.Compressez le Cisco CallManager et les fichiers de suivi SDL ensemble.Envoyez et copiez ce fichier zip tandis que vous attendez le contact.Compressez les logs de perfmon ensemble.Envoyez et copiez ce fichier zip tandis que vous attendez le contact.Compressez les entrées de journal d'événements ensemble.Envoyez et copiez ce fichier zip tandis que vous attendez le contact.
6. Après que vous ayez recueilli tous les suivis et vous connectiez, compressez les fichiers et envoyez le fichier zip à attach@cisco.com. Fournissez le numéro de dossier dans le champ objet de courrier électronique.

Informations connexes

- Panne de service Cisco CallManager
- Dépannage des pannes Cisco CallManager
- Assistance technique concernant la technologie vocale
- Assistance concernant les produits vocaux et de communications unifiées
- Dépannage des problèmes de téléphonie IP Cisco
- Support et documentation techniques - Cisco Systems