StarOS機能のクラッシュのトラブルシューティング

内容

はじめに

このドキュメントでは、StarOsファシリティのクラッシュを見つけてトラブルシューティングする方法について説明します。

概要

場合によっては、システムロジックに障害が発生し、正常な機能を復元するためにソフトウェアタスクが再起動することがあります。これにより、プロセスがクラッシュする可能性があります。タスクファシリティのクラッシュはStarOSで頻繁に報告され、クラッシュの根本原因に基づいて必要なアクションを実行できます。ノード上のクラッシュを特定するには、次のCLIコマンドを使用できます。

```
 ******** show crash list *******
Saturday April 15 05:05:56 SAST 2023
=== ==================== ======== ========== =============== ======================
#           Time          Process  Card/CPU/        SW          HW_SER_NUM
                                      PID         VERSION       CF / Crash Card
=== ==================== ======== ========== =============== ======================

1   2022-Dec-02+14:08:46 confdmgr 02/0/19342 21.26.13        NA
2   2022-Dec-02+14:48:08 confdmgr 02/0/31546 21.26.13        NA
3   2022-Dec-04+19:10:50 sessmgr  03/0/12321 21.26.13        NA
4   2022-Dec-21+03:34:13 sessmgr  04/0/12586 21.26.13        NA
```

同様のクラッシュは、1つのレコードに統合されます。レコードには、このクラッシュタイプが発生した回数が表示されます。

```
********************* CRASH #02 ***********************
SW Version          : 21.26.13
Similar Crash Count : 33 >>>>
Time of First Crash : 2022-Dec-02+14:10:05

Assertion failure at confdmgr/src/confdmgr_fsm.c:870
  Note: State machine failure, state = 3
  Function: confdmgr_fsm_state_wait_p0_handler()
  Expression: 0
  Code: CRASH
  Proclet: confdmgr (f=1900,i=0)
  Process: card=2 cpu=0 arch=X pid=31546 argv0=confdmgr
```

インshow snmp trap history verbose次の出力は、一部のプロセスがクラッシュしたことを示しています。

```
Fri Dec 26 08:32:20 2014 Internal trap notification 73 (ManagerFailure) facility sessmgr
instance 188 card 7 cpu 0
Fri Dec 26 08:32:20 2014 Internal trap notification 150 (TaskFailed) facility sessmgr instance
188 on card 7 cpu 0
Fri Dec 26 08:32:23 2014 Internal trap notification 1099 (ManagerRestart) facility sessmgr
instance 139 card 4 cpu 1
Fri Dec 26 08:32:23 2014 Internal trap notification 151 (TaskRestart) facility sessmgr
instance 139 on card 4 cpu 1
```

クラッシュのシナリオ

クラッシュの原因は複数ある可能性があります。

1.さまざまなコールフローシナリオ

2.メモリの問題

3.設定の問題

4.ハードウェア障害

クラッシュの原因

StarOSには複数のタスク機能があり、機能に基づいて機能が異なります。問題のある状態に陥りそうな入力があると、機能はクラッシュし、エラー状態から回復します。

さまざまなタイプのクラッシュ

1.アサーションの失敗：

```
********************* CRASH #22 ***********************
```

```
SW Version          : 21.26.13
Similar Crash Count : 33
Time of First Crash : 2023-Apr-12+22:40:01

Assertion failure at sess/smgr/sessmgr_snx.c:9568 >>>>
  Function: sessmgr_snx_send_drop_call()
  Expression: result == SN_STATUS_SUCCESS
  Proclet: sessmgr (f=87000,i=261)
  Process: card=5 cpu=0 arch=X pid=12724 cpu=~0% argv0=sessmgr
```

2.セグメンテーション障害：

```
******************** CRASH #69 ***********************
SW Version : 21.13.3
Similar Crash Count : 2
Time of First Crash : 2019-Nov-25+07:53:54
Fatal Signal 11: Segmentation fault >>>>
Faulty address: 0x7ff6b4801036
Signal from: kernel
Signal detail: address not mapped to object
Process: card=8 cpu=1 arch=X pid=7316 argv0=vpp
Crash time: 2020-Feb-11+04:04:23 UTC
Build_number:
```

3.致命的なシグナル：

```
******************** CRASH #01 ***********************
SW Version          : 21.23.12
Similar Crash Count : 2
Time of First Crash : 2023-Jan-27+05:22:46

Fatal Signal 11: 11  >>>>>
  PC: [04be6859/X] sessmgr_pgw_create_bearers()
  Faulty address: 0x297116e4
  Signal from: kernel
  Signal detail: address not mapped to object
  Process: card=9 cpu=1 arch=X pid=10383 cpu=~8% argv0=sessmgr
```

初期ログ要件

クラッシュログは、クラッシュイベント情報の貴重なソースとして役立ちます。ソフトウェアクラッシュが発生すると、StarOSはクラッシュの原因の特定に役立つ関連データをキャプチャして保存します。この情報は、システムメモリに保存することも、ネットワークサーバに転送して保存することもできます。

コアファイルまたはミニコアファイル：コアファイルはクラッシュが発生したPIDに対応することに注意してください。コアファイルの名前は、「crash-<card no>-<cpu>-<pid>-<unixtime>-

core」の形式で指定されます。この情報は「show crash list」コマンドの出力で確認できます。

Minicoreファイル：このファイルには、現在のスタックトレース、過去のプロファイラサンプル、過去のメモリアクティビティサンプル、および独自のファイル形式のその他のバンドルデータなど、失敗したタスクに関する情報が含まれています。

コアダンプ（またはフルコア）：コアダンプは、クラッシュが発生した直後のプロセスの完全なメモリダンプを提供します。このメモリダンプは、多くの場合、ソフトウェアクラッシュの根本原因を特定するために不可欠です。

クラッシュシグニチャ：クラッシュシグニチャは、共有されているShow Support Details(SSD)またはその他の関連ソースから確認できます。

```
 ******** show crash list *******
Saturday April 15 05:05:56 SAST 2023
=== ==================== ======== ========== =============== ======================
#            Time         Process  Card/CPU/        SW          HW_SER_NUM
                                      PID         VERSION       CF / Crash Card
=== ==================== ======== ========== =============== ======================

1   2022-Dec-02+14:08:46 confdmgr 02/0/19342 21.26.13        NA
2   2022-Dec-02+14:48:08 confdmgr 02/0/31546 21.26.13        NA
3   2022-Dec-04+19:10:50 sessmgr  03/0/12321 21.26.13        NA
```

クラッシュ1のシグニチャを確認するには、CRASH #01を使用してSSDで検索するか、CLIでshow crash number 1を使用します。

```
From SSD

********************* CRASH #01 ***********************
SW Version          : 21.26.13
Similar Crash Count : 1
Time of First Crash : 2022-Dec-02+14:08:46

Assertion failure at confdmgr/src/confdmgr_fsm.c:758
  Note: State machine failure, state = 5
  Function: confdmgr_fsm_state_wait_p1_handler()
  Expression: 0
  Code: CRASH

Using CLI

[local]abc# show  crash number 1
Friday June 09 06:41:53 CDT 2023
********************* CRASH #01 ***********************
SW Version          : 21.12.20.77760
Similar Crash Count : 1
Time of First Crash : 2021-Mar-31+15:58:06

Fatal Signal 6: Aborted
  PC: [ffffe430/X] __kernel_vsyscall()
  Note: User-initiated state dump w/core.
```

```
Signal from: sitmain pid=6999 uid=0
Process: card=9 cpu=0 arch=X pid=9495 cpu=~0% ar
```

問題が発生した特定のタイムスタンプの間にShow Support Details(SSD)とsyslogを調べます。

分析手順

1.クラッシュスタック/シグニチャをチェックし、その特定のクラッシュシグニチャにバグがないかチェックする必要があります。

2. corefile/minicoreを解析してバックトレースを分析し、ファシリティがクラッシュした機能のヒントを取得する必要があります。

3.コアファイルのデバッグが完了したら、ソフトウェア不具合を使用して症状を確認し、同様のクラッシュシグニチャとバックトレースに対する既存のソフトウェア不具合があるかどうかを確認します。

セッション回復

StarOsソフトウェアは、予測される状況/イベントと予測されない状況/イベントの両方を処理するように設計されています。シスコは完璧なソフトウェアを提供するよう努めていますが、間違いは避けられず、クラッシュする可能性があります。このために、セッション リカバリ機能は非常に重要です。

セッション復旧機能は、システム内のハードウェアまたはソフトウェアの障害が発生した場合に、完全に接続されたユーザセッションの切断を妨げる加入者セッション情報のシームレスなフェールオーバーと再構築を提供します。セッションのリカバリは、システム内の主要なソフトウェアプロセス（たとえば、セッションマネージャやAAAマネージャ）をミラーリングすることによって実行されます。ミラー化されたプロセスはアイドル状態（スタンバイ・モード）のままになり、ソフトウェア障害（セッション・マネージャ・タスクの中断など）が発生した場合に必要になるまで、プロセスは処理を実行しません。

demuxタスク、AAAマネージャ、VPNマネージャなどのタスクには、特に加入者情報を処理するための自動回復メカニズムが組み込まれています。セッションリカバリとは主に、sessmgrタスクまたはカードレベルの障害が発生し、コールを失うことなくセッションをリカバリする必要があるシナリオを指します。

- システムでは、スタンバイセッションマネージャが各処理カードでアクティブになり、障害が発生した場合に迅速にプライマリセッションマネージャを引き継ぐことができます。次に、新しいスタンバイsessmgrが処理カード上に作成されます。
- sessmgrプロセスが予期せず失敗すると、スタンバイsessmgrはaaamgr（AAAマネージャ）からバックアップ情報を取得し、セッションを再構築します。
- aaamgrに障害が発生すると、スタンバイaaamgrはsessmgrに照会して、関連する加入者情報を同期します。
- demux-mgrで障害が発生すると、すべてのsessmgrを照会し、コール分散情報のデータベースを再構築します。

- カードレベルの冗長性を確保するために、1枚の処理カードがスタンバイとして機能し、ハードウェアまたはソフトウェアの障害時に引き継ぐことができます。スタンバイカードは、別のカードで実行されているピアaamgrからセッションを回復します。

```
 ******** show session recovery status verbose *******
Saturday April 15 05:11:17 SAST 2023
Session Recovery Status:
  Overall Status        : Ready For Recovery  >>>>
  Last Status Update    : 5 seconds ago

              ----sessmgr---  ----aaamgr----  demux
 cpu state    active standby  active standby  active  status
---- -------  ------ -------  ------ -------  ------  -------------------------
 3/0 Active   40     1        40     1        0       Good
 4/0 Active   40     1        40     1        0       Good
 5/0 Active   40     1        40     1        0       Good
 6/0 Active   40     1        40     1        0       Good
 7/0 Active   0      0        0      0        10      Good (Demux)
 8/0 Active   40     1        40     1        0       Good
 9/0 Active   40     1        40     1        0       Good
10/0 Active   40     1        40     1        0       Good
11/0 Standby  0      40       0      40       0       Good
```

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。