



## セッションリカバリ

堅牢なハードウェアフェールオーバーと冗長性保護により、システム上のハードウェアやソフトウェアの障害を迅速に修正できます。ただし、さまざまな理由でソフトウェアの障害が発生する可能性があり、ほとんどの場合、何の前触れ也没有ありません。

この章では、ハードウェアまたはソフトウェアに障害が発生した場合に、サブスクライバ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行うセッションリカバリ機能について説明します。



**重要** セッションリカバリは、シスコのライセンス対象機能です。別の機能ライセンスが必要になる場合があります。特定のライセンス要件の詳細については、シスコのアカウント担当者にお問い合わせください。ライセンスのインストールと確認の詳細については、「ソフトウェア管理操作」の「ライセンスキーの管理」の項を参照してください。

この章は、次の項で構成されています。

- [セッションリカバリの仕組み](#) (1 ページ)
- [ASR 5500 の追加ハードウェア要件](#) (5 ページ)
- [セッションリカバリをサポートするためのシステムの設定](#) (6 ページ)
- [制御タスクの統計情報のリカバリ](#) (11 ページ)

## セッションリカバリの仕組み

この項では、この機能の実装方法とリカバリプロセスについての概要を示します。

セッションリカバリ機能は、システム内のハードウェアまたはソフトウェアに障害が発生した場合に、サブスクライバ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行い、完全に接続されたユーザーセッションが切断されるのを防ぎます。

セッションリカバリは、システム内の重要なソフトウェアプロセス（セッションマネージャや AAA マネージャなど）をミラーリングすることによって実行されます。これらの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）を維持し、ソフトウェア障害が発生した場

合（セッションマネージャのタスクが中止された場合など）に必要なまでいかなる処理も実行しません。

システムは、使用されているアクティブな制御プロセッサ（CP）ごとに、「スタンバイモード」セッションと AAA マネージャの新しいインスタンスを生成します。これらのミラー化されたプロセスには、メモリリソースと処理リソースの両方が必要です。つまり、この機能を有効にするには、追加のハードウェアが必要になる場合があります（[ASR 5500 の追加ハードウェア要件（5 ページ）](#)を参照）。

VPN マネージャなどのその他の重要なシステムレベルのソフトウェアタスクは、物理的に分離されたパケット処理カード上で実行され、二重のソフトウェア障害（セッションマネージャと VPN マネージャが同じカード上で同時に失敗するなど）が発生しないようにします。VPN マネージャプロセスをホストするパケット処理カードはアクティブモードであり、セッションリカバリが有効になったときのみ使用する目的でオペレーティングシステムによって予約されています。

セッションリカバリには 2 つのモードがあります。

- タスクリカバリモード**：スタンバイパケット処理カード上でリソースを使用することなく、1 つ以上のセッションマネージャの障害が発生し、回復されます。このモードでは、アクティブなパケット処理カード上で実行しているミラー化された「スタンバイモード」のセッションマネージャのタスクを使用してリカバリが実行されます。「スタンバイモード」のタスクは名前が変更され、アクティブになった後、AAA マネージャなどの他のタスクの情報を使用して入力されます。タスクが失敗した場合は、限られたサブスクリイバが影響を受け、タスクが再開するまで停止します。
- フルパケット処理カードリカバリモード**：パケット処理カードのハードウェア障害が発生した場合、または計画されていたパケット処理カードの移行が失敗した場合に使用します。このモードでは、スタンバイパケット処理カードがアクティブになり、新たにアクティブ化されたパケット処理カード上の「スタンバイモード」のセッションマネージャと AAA マネージャタスクによってセッションリカバリが実行されます。

セッション/コールの状態情報は、各 AAA マネージャのタスクとセッションマネージャのタスクがペアになっているため、ピア AAA マネージャのタスクに保存されます。これらのペアは、物理的に異なるパケット処理カードで開始され、タスクのリカバリを確実にします。

セッションリカバリが適切に動作しない場合があります。これには次が含まれます。

- セッションリカバリ動作中に、新たなソフトウェアまたはハードウェアの障害が発生した場合。たとえば、AAA マネージャは、そこに含まれていた状態情報が、新たにアクティブになったセッションマネージャのタスクの入力に使用されている間に失敗します。
- セッションリカバリをサポートするためのハードウェアのリソース（パケット処理カードのメモリや制御プロセッサ）が不足している場合。



**重要** セッションリカバリ動作の後、マネージャごと（AAA マネージャ、セッションマネージャなど）に収集および保持された統計情報など、一部の統計情報は、通常は回復されません。アカウントリングおよび課金関連の情報のみがチェックポイントングされ、回復します。

セッションリカバリは、次の機能で使用できます。

- L2TP LAC サポートを必要とするセッション（HA や GGSN セッションの上で再生成された PPP を除く）
- ASR 5500 のみ：簡易 IP、モバイル IP、およびプロキシモバイル IP をサポートしているクラウド RP PDSN サービス
- ASR 5500 のみ：eHRPD サービス（進化した高レートパケットデータ）
- ASR 5500 のみ：ePDG サービス（進化したパケットデータゲートウェイ）
- ASR 5000 のみ：eWAG サービス（拡張ワイヤレスアクセスゲートウェイ）
- IPv4 コンテキストと PPP PDP コンテキストの GGSN サービス
- ユーザーごとのレイヤ 3 トンネルの有無にかかわらず、モバイル IP および/またはプロキシモバイル IP セッションタイプをサポートしている HA サービス
- ASR 5500 のみ：HNB-GW : IuH を介した HNB セッション
- ASR 5500 のみ：HNB-GW : IuPS と IuCS を介した HNB-CN セッション
- ASR 5500 のみ：HNB-GW : SeGW セッションの IPSec トンネル
- ASR 5500 のみ：IPv4 用の HSGW サービス
- IPCF（インテリジェントポリシー制御機能）
- ASR 5500 のみ：IPSG 専用システム（IP サービスゲートウェイ）
- LNS セッションタイプ（L2TP ネットワークサーバー）
- MME（モビリティ マネージメント エンティティ）
- ASR 5500 のみ：NEMO（ネットワークモビリティ）
- IPv4 用の P-GW サービス
- ASR 5000 のみ：PDG/TTG（パケットデータゲートウェイ/トンネル終端ゲートウェイ）
- ASR 5500 のみ：PDIF（パケットデータ インターワーキング機能）
- 簡易 IP、モバイル IP、およびプロキシモバイル IP をサポートしている PDSN サービス
- S-GW（サービングゲートウェイ）
- SGSN（サービング GPRS サポートノード）サービス
- ASR 5000 と VPC-DI : IPv6 と IPv4IPv6（デュアル）の PDP セッションリカバリは、3G サービスと 2G サービスでサポートされます。
- SaMOG（GTP を介した S2a モビリティ）ゲートウェイ（CGW と MRME）
- ASR 5500 のみ：SAE-GW（System Architecture Evolution ゲートウェイ）
- SCM（サービス制御マネージャ）

- ASR 5500 のみ：IPv4 コンテキストと PPP PDP コンテキスト用の SGSN サービス（3G サービスと 2.5 G サービス）

セッションリカバリは、次の機能ではサポートされていません。

- 接続先ベースのアカウントングリカバリ
- GGSN ネットワークによって開始された接続
- 2 つ以上のサービスインスタンスを使用する GGSN セッション
- IPSec 統合を使用した MIP/L2TP
- 複数の同時バインディングを使用した MIP セッション
- L2TP を使用したモバイル IP セッション
- 複数の MIP セッション
- RAB リカバリ




---

**重要** その他の可能なセッションリカバリおよびシャード間セッションリカバリ（ICSR）のサポートの制限については、常に個々の製品のアドミニストレーションガイドを参照してください。

---

セッションリカバリが発生すると、システムは次のサブスクリバ情報を再構築します。

- 正しいコール動作を維持するために必要なデータと制御状態情報。
- サブスクリバデータの最小の統計情報セット。アカウントング情報を確実に保持するために必要です。
- コール期間、絶対時間などのさまざまなタイマー値を回復するためのベストエフォート型の試行。
- アイドル時間タイマーをゼロにリセットし、再登録タイマーを HA セッションの最大値にリセットして、セッションリカバリに対してより控えめなアプローチをとります。

また、セッションリカバリは、ソフトウェアパッチのアップグレードアクティビティにも役立ちます。ソフトウェアパッチプロセス時に有効になっている場合、セッションリカバリでは、アクティブなパケット処理カード上の既存のセッションが保持されます。詳細については、「ソフトウェア管理操作」の章の「動的なソフトウェアパッチ更新の実行」を参照してください。




---

**重要** 部分的に接続されたコール（HA 認証が保留されていても、AAA サーバーによってまだ応答確認されていないセッションなど）は、障害が発生しても回復されません。

---



- (注) クリティカルなタスクが失敗すると、StarOSが再起動されます。カーネルの障害、ハイパーバイザの障害、またはハードウェアの障害によって、VMが再起動するか、またはオフラインになります。これらのタイプの障害の解決策として、2つのVPC-DI間または2つのVPC-SI間でICSRを使用することを推奨します。

## ASR 5500 の追加ハードウェア要件

セッションリカバリでは、メモリ、制御プロセッサ、NPU処理能力などの多くのハードウェアリソースが必要なため、この機能を完全にサポートするのに十分なリソースを確保するために、ハードウェアの追加が必要になる場合があります。



- 重要** この機能を使用するには、個々のシャーシに最低4つのパケット処理カード（アクティブが3つとスタンバイが1つ）が必要です。

パケット処理カードの移行中にハードウェア障害が発生した場合に、セッションを完全に回復できるようにするには、少なくとも3つのアクティブなパケット処理カードと2つのスタンバイパケット処理カードを配置する必要があります。

ネットワーク設計とキャパシティプランニングを効果的に行えるよう、次の点を考慮してください。

- 加入者キャパシティは、ハードウェアの設定によって減少します。完全構成のシャーシでは、最小構成のシャーシと比べて、加入者キャパシティの減少はわずかです。
- 制御トランザクション処理キャパシティが削減される量。
- 加入者データのスループットの低下。
- 失敗したソフトウェアタスクのリカバリ時間。
- 障害が発生したパケット処理カードのリカバリ時間。

パケット処理カードの移行は、移行中に必要になる可能性のあるハードウェアリソース（メモリ、プロセッサなど）が使用できないため、一時的にセッションリカバリに影響を与える可能性があります。この状態を回避するには、少なくとも2つのスタンバイパケット処理カードを設定する必要があります。



- (注)
- メモリの削減により、新しいカードのセッションマネージャのメモリ不足が発生し、セッションマネージャの一部が警告またはオーバーの状態になります。セッションマネージャによって割り当てられたメモリは、移行によるアドレスの再指定後は増大しません。
  - 各 Session Manager プロセスの共有メモリが移行後に専用メモリになるため、システムで使用可能なメモリの合計は、カードの移行時に減少します。これにより、複数のコピーが作成されるため、より多くのメモリが占有されます。したがって、全構成がロードされた後（day-1 設定後）、少なくとも 4～5 GB の使用可能なメモリがあることが推奨されます。使用可能なメモリがない場合、共有メモリから専用メモリへの変換によりメモリ使用量が増加するため、カードの移行後に使用可能なメモリ量が減少します。

## セッションリカバリをサポートするためのシステムの設定

次に、現在、動作している運用システム（着信コールを受け入れることができる）か、動作していないシステム（実稼働ネットワークに含まれておらず、そのためライブサブスクライバ/カスタマーのデータを処理していない）のいずれかに対してセッションリカバリ機能を設定する手順を示します。



- 重要** 機能使用キーがある場合でも、デフォルトでは、セッションリカバリ機能はシステム上で無効になっています。

### セッションリカバリの有効化

前述したように、アウトオブサービス（OOS）でコンテキストがまだ設定されていないシステム、または現在コールを処理できるインサービスシステムで、セッションリカバリを有効にすることができます。ただし、システムがインサービスである場合は、セッションリカバリ機能が有効になる前に再起動する必要があります。

#### アウトオブサービスシステムでのセッションリカバリの有効化

次に、コンテキストが設定されていないシステムの場合の手順を示します。

アウトオブサービスシステムでセッションリカバリ機能を有効にするには、次の手順を実行します。この手順は、Exec モードプロンプトで開始することを前提としています。

## 手順

**ステップ 1** Exec モードのプロンプトで、**show license info** コマンドを実行し、システム上のセッションと機能の使用ライセンスを介してセッションリカバリ機能が有効になっていることを確認します。

セッションリカバリ機能の現在のステータスが [Disabled] になっている場合は、ライセンスキーをシステムにインストールするまで、この機能を有効にすることはできません。

**ステップ 2** セッションリカバリを有効にするには、次の設定例を使用します。

```
configure
require session recovery
end
```

(注) このコマンドを設定した後で、設定を保存してからシャーシをリロードし、コマンドを有効にする必要があります。設定ファイルを保存してシャーシをリロードする方法については、使用している展開の『System Administration Guide』を参照してください。

**ステップ 3** 「設定の確認と保存」の説明に従って、設定を保存します。

起動すると、システムはセッションリカバリを有効にし、ミラーリングされた「スタンバイモード」のタスクをすべて作成し、パケット処理カードの予約やその他の操作を自動的に実行します。

**ステップ 4** システムが設定され、インサービスになったら、この機能をサポートするための準備がシステムに整っていることを管理します。次の説明にしたがってください。 [セッションリカバリ ステータスの表示 \(8 ページ\)](#)

## インサービスシステムでのセッションリカバリの有効化

すでに保存されている設定があるシステムでセッションリカバリを有効にすると、セッションリカバリコマンドは、設定ファイル内のサービス構成コマンドの前に自動的に配置されます。

インサービスシステムでセッションリカバリ機能を有効にするには、次の手順に従います。この手順は、Exec モードプロンプトで開始することを前提としています。

## 手順

**ステップ 1** Exec モードプロンプトで **show license info** コマンドを実行して、セッションと機能の使用ライセンスを使用して、セッションリカバリ機能が有効になっていることを確認します。

セッションリカバリ機能の現在のステータスが [Disabled] になっている場合は、ライセンスキーをシステムにインストールするまで、この機能を有効にすることはできません。

**ステップ 2** セッションリカバリを有効にするには、次の設定例を使用します。

```
configure
  require session recovery
end
```

この機能は、システムが再起動されるまでは有効になりません。

**ステップ3** 「設定の確認と保存」の説明に従って、設定を保存します。

**ステップ4** **reload** コマンドを入力して、システムの再起動を実行します。

次のプロンプトが表示されます。

```
Are you sure? [Yes|No]:
```

**Yes** と入力して、システム再起動の実行を確認します。

再起動すると、システムはセッションリカバリを有効にし、ミラーリングされた「スタンバイモード」のタスクをすべて作成し、パケット処理カードの予約やその他の操作を自動的に実行します。

**ステップ5** システムが再起動したら、システムがこの機能をサポートするように準備されていることを確認する必要があります。次の説明に従ってください。 [セッションリカバリ ステータスの表示 \(8 ページ\)](#)

上級ユーザーはテキストユーザーなどの手段を使用して、**require session recovery** コマンドシンタックスを既存の設定ファイルに挿入してから、設定ファイルを手動で適用することもできます。このコマンドが既存の設定ファイルの最初の数行の間に配置されるようにする際は注意してください。これは、ローカル以外のコンテキストの作成よりも前に表示される必要があります。

## セッションリカバリ機能の無効化

システムでセッションリカバリ機能を無効にするには、グローバル構成モードのプロンプトから **no require session recovery** コマンドを入力します。



**重要** このコマンドがインサーブシステムで発行された場合は、**reload** コマンドを発行してシステムを再起動する必要があります。

## セッションリカバリ ステータスの表示

システムがセッションリカバリを実行できるかどうかを確認するには、イネーブルにして、Exec モードプロンプトから **show session recovery status verbose** コマンドを入力します。

このコマンドの出力は、次に示す例のようになります。

```
[local]host_name# show session recovery status
Session Recovery Status:
  Overall Status           : SESSMGR Not Ready For Recovery
  Last Status Update      : 1 second ago
```

```
[local]host_name# show session recovery status
Session Recovery Status:
```



```

Overall Status           : Ready For Recovery
Last Status Update      : 8 seconds ago

[local]host_name# show session recovery status verbose
Session Recovery Status:
Overall Status           : Ready For Recovery
Last Status Update      : 2 seconds ago

-----sessmgr-----
cpu state   active   standby
-----
1/1 Active   2         1
1/2 Active   1         1
1/3 Active   1         1
2/1 Active   1         1
2/2 Active   1         1
2/3 Active   2         1
3/0 Active   0         0
3/2 Active   0         0
4/1 Standby  0         2
4/2 Standby  0         1
4/3 Standby  0         2

-----aaamgr-----
active   standby
-----
1         1
0         0
3         1
1         1
0         0
3         1
0         0
0         0
0         1
0         1
0         0
0         0
0         3

demux
active
-----
0
0
0
0
0
0
1
1
0
0
0
0

status
-----
Good
Good
Good
Good
Good
Good
Good (Demux)
Good (Demux)
Good
Good
Good
Good

[local]host_name#

```

## セッション情報回復の確認

サブスライバごとのセッション情報を使用して、セッションリカバリステータスの変更を表示できます。**show subscriber debug-info** コマンドの出力の [Redundancy Status] というフィールドには、セッションが回復したかどうかや元の情報であるかどうかを示されます。このフィールドには、次の2つの有効な出力があります。

- **Original** : これは、すべてのイベント状態と時刻情報を含む元のセッション情報であることを示します。
- **Recreated Session** : セッションリカバリ操作中にこのセッションが再構築されたことを示します。

このコマンドは、セッションリカバリ操作が実行される前または後に実行でき、特定のセッションに関連する情報を表示します。

セッション状態情報とセッションリカバリステータスを表示するには、次のコマンドを入力します。

```
[local]host_name# show subscriber debug-info { callid id | msid id | username name }
```

キーワード/変数	説明
<b>callid id</b>	<i>id</i> で指定されたコールのサブスライバ情報を表示します。コール ID は 8 バイトの 16 進数です。

キーワード/変数	説明
<b>msid id</b>	MSID によって識別されたモバイルユーザーの情報を、IMSI、MIN、または RMI として指定された 7～16 桁で表示します。ワイルドカード文字の \$ と * は使用できます。ワイルドカードの * は複数の文字と一致し、ワイルドカードの \$ は 1 文字と一致します。ワイルドカード文字がワイルドカードとして解釈されないようにするには、一重引用符 (') で囲みます。たとえば、'\$' のように指定します。
<b>username name</b>	以前に設定したユーザー名によって識別されるサブスクライバの接続に関する情報を表示します。 <i>name</i> は、1 から 127 文字の長さの一連の英数字および/またはワイルドカード文字 (「\$」と「*」) です。ワイルドカードの * は複数の文字と一致し、ワイルドカードの \$ は 1 文字と一致します。ワイルドカード文字がワイルドカードとして解釈されないようにするには、一重引用符 (') で囲みます。たとえば、'\$' のように指定します。

次に、このコマンドのセッションリカバリ操作が実行される前と後の両方の出力例を示します。わかりやすくするために、この例の [Redundancy Status] フィールドを太字で示しています。

```

username: user1                callid: 01callb1                msid: 0000100003
Card/Cpu: 4/2
Sessmgr Instance: 7
Primary callline:
Redundancy Status: Original Session
  Checkpoints      Attempts      Success      Last-Attempt      Last-Success
  Full:            69            68            29800ms           29800ms
  Micro:           206           206           20100ms           20100ms
Current state: SMGR_STATE_CONNECTED
FSM Event trace:
  State            Event
  SMGR_STATE_OPEN  SMGR_EVT_NEWCALL
  SMGR_STATE_NEWCALL_ARRIVED SMGR_EVT_ANSWER_CALL
  SMGR_STATE_NEWCALL_ANSWERED SMGR_EVT_LINE_CONNECTED
  SMGR_STATE_LINE_CONNECTED SMGR_EVT_LINK_CONTROL_UP
  SMGR_STATE_LINE_CONNECTED SMGR_EVT_AUTH_REQ
  SMGR_STATE_LINE_CONNECTED SMGR_EVT_IPADDR_ALLOC_SUCCESS
  SMGR_STATE_LINE_CONNECTED SMGR_EVT_AUTH_SUCCESS
  SMGR_STATE_LINE_CONNECTED SMGR_EVT_UPDATE_SESS_CONFIG
  SMGR_STATE_LINE_CONNECTED SMGR_EVT_LOWER_LAYER_UP
Data Reorder statistics
Total timer expiry: 0          Total flush (tmr expiry): 0
Total no buffers: 0          Total flush (no buffers): 0
Total flush (queue full): 0  Total flush (out of range): 0
Total flush (svc change): 0  Total out-of-seq pkt drop: 0
Total out-of-seq arrived: 0
IPv4 Reassembly Statistics:
  Success: 0          In Progress: 0
  Failure (timeout): 0  Failure (no buffers): 0
  Failure (other reasons): 0
Redirected Session Entries:      Allowed:
2000      Current: 0
          Added: 0          Deleted:
          0
          Revoked for use by different subscriber: 0
Peer callline:

```

```

Redundancy Status: Recovered Session
Checkpoints      Attempts      Success      Last-Attempt      Last-Success
  Full:          0              0              0ms                0ms
  Micro:         0              0              0ms                0ms
Current state: SMGR_STATE_CONNECTED
FSM Event trace:
  State                                          Event
SMGR_STATE_LINE_CONNECTED                    SMGR_EVT_LOWER_LAYER_UP
SMGR_STATE_CONNECTED                         SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED                         SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED                         SMGR_EVT_REQ_SUB_SESSION
SMGR_STATE_CONNECTED                         SMGR_EVT_RSP_SUB_SESSION
SMGR_STATE_CONNECTED                         SMGR_EVT_ADD_SUB_SESSION
SMGR_STATE_CONNECTED                         SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED                         SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED                         SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED                         SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED                         SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED                         SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED                         SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED                         SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED                         SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED                         SMGR_EVT_AUTH_SUCCESS
Data Reorder statistics
  Total timer expiry:          0          Total flush (tmr expiry): 0
  Total no buffers:           0          Total flush (no buffers): 0
  Total flush (queue full):   0          Total flush (out of range):0
  Total flush (svc change):   0          Total out-of-seq pkt drop: 0
  Total out-of-seq arrived:  0
IPv4 Reassembly Statistics:
  Success:                     0          In Progress:              0
  Failure (timeout):           0          Failure (no buffers):     0
  Failure (other reasons):     0
Redirected Session Entries:
  Allowed:                     2000       Current:                  0
  Added:                       0          Deleted:                  0
  Revoked for use by different subscriber: 0

```

## 制御タスクの統計情報のリカバリ

リカバリ制御タスク（RCT）の統計情報には、次の情報が表示されます。

- リカバリアクションの実行：移行、シャットダウン、スイッチオーバー
- イベントのタイプ：計画済みまたは未計画
- カードからカードへ：スロット番号
- 開始時刻：YYYY-MMM-DD+hh:mm:ss.sss
- 期間：秒
- カード障害のデバイス（CPU<sub>n</sub> など）
- カード障害の理由
- カードが使用可能な状態であるか、障害はない
- リカバリアクションのステータス：成功または失敗の理由
- リカバリアクションが失敗した場合、失敗したタイムスタンプ
- リカバリアクションが失敗した場合、失敗したタスクファシリティ名
- リカバリアクションが失敗した場合、失敗したインスタンス番号

## show rct stats コマンド

Exec モードの **show rct stats** コマンドでは、次のシンタックスが採用されています。

```
[local]host_name# show rct stats [verbose]
```

**verbose** キーワードを指定しないと、次の例に示すようにサマリー出力が表示されます。

RCT stats details (Last 1 Actions)

#	Action	Type	From	To	Start Time	Duration	Status
1	Migration(st)	Planned	2	1	2016-Jul-12+13:12:21.865	0.003 sec	Success

RCT stats summary

```
-----
Migrations = 0
  Management Card: 0 Average time: 0.000 sec
  Packet Card : 1 Average time: 0.006 sec
Switchovers = 1, Average time - 25.855 sec
```

**verbose** キーワードを指定すると、[show rct stats verbose の出力例 \(12 ページ\)](#) に示す詳細な統計情報が提供されます。

## show rct stats verbose の出力例

```
[local]host_name# show rct stats verbose
```

RCT stats Details (Last 5 Actions)

```
Stats 1:
  Action      : Migration
  Type       : Planned
  From       : 5
  To        : 6
  Start Time : 2017-Apr-04+03:02:00.132
  Failure Reason : CPU_CRITICAL_TASK_FAILURE
  Failure Device : CPU_0
  Is Card Usable : Yes
  Recovery Status : Success
  Facility     : N.A
  Instance    : N.A
  Duration    : 066.050 sec
  Graceful    : Enabled
  Recovered [1] : [f:sessmgr, i:6, cpu:50, pid:13170]
  Recovered [2] : [f:sessmgr, i:3, cpu:50, pid:13167]
```

RCT stats Details (Last 5 Actions)

```
Stats 2:
  Action      : Shutdown
  From       : 12
  To        : 13
  Start Time : 2017-Apr-04+03:02:10.100
  Is Card Usable : Yes
  Failure Reason : NPU_LC_CONNECT_TOP_FAIL
  Failure Device : PAC_LC_CONNECT_HARDWARE
  Recovery Status : Success
  Facility     : N.A
  Instance    : N.A
```

```
Duration      : 002.901 sec
Graceful      : Enabled
  Recovered [1] : [f:sessmgr, i:6, cpu:50, pid:13170]
  Recovered [2] : [f:sessmgr, i:3, cpu:50, pid:13167]

Stats 3:
Action        : Migration
From          : 7
To            : 11
Start Time    : 2017-Apr-04+03:03:40.120
Is Card Usable : Yes
Failure Reason : N.A.
Failure Device : N.A
Recovery Status : Success
Facility      : N.A
Instance      : N.A
Duration      : 003.423 sec
Graceful      : Enabled
  Recovered [1] : [f:sessmgr, i:6, cpu:50, pid:13170]
  Recovered [2] : [f:sessmgr, i:3, cpu:50, pid:13167]

Stats 4:
Action        : Migration
From          : 7
To            : 11
Start Time    : 2017-Apr-04+03:03:41.256
Is Card Usable : Yes
Failure Reason : N.A.
Failure Device : N.A
Recovery Status : TASK_MIGRATION_FAIL_PREMIGRATE
Facility      : vpnmgr
Instance      : 13
Duration      : 005.222 sec
Graceful      : Enabled
  Recovered [1] : [f:sessmgr, i:6, cpu:50, pid:13170]
  Recovered [2] : [f:sessmgr, i:3, cpu:50, pid:13167]

Stats 5:
Action        : Migration
From          : 6
To            : 7
Start Time    : 2017-Apr-04+04:18:30.106
Is Card Usable : Yes
Failure Reason : N.A.
Failure Device : N.A
Recovery Status : TASK_MIGRATION_FAIL_RENAME
Facility      : sessmgr
Instance      : 63
Duration      : 004.134 sec
Graceful      : Enabled
  Recovered [1] : [f:sessmgr, i:6, cpu:50, pid:13170]
  Recovered [2] : [f:sessmgr, i:3, cpu:50, pid:13167]

RCT stats Summary
-----
Migrations =      3, Average time = 4.260 sec
Switchovers =      0
```



## 翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。