

# Oracle® Solaris 11.1 での仮想ネットワークの使用

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

#### U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する際、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD, Opteron, AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

# 目次

---

はじめに .....	5
<b>1 Oracle Solaris</b> でのネットワーク仮想化およびリソース管理 .....	7
ネットワーク仮想化の概要 .....	7
ネットワーク仮想化のコンポーネント .....	8
仮想ネットワークを実装すべきユーザー .....	11
仮想化コンポーネントを構成するためのコマンド .....	11
ネットワークリソースの管理の概要 .....	12
リソース制御のためのデータリンクプロパティ .....	12
フローを使用することによるネットワークリソースの管理 .....	13
ネットワークリソースの管理のためのコマンド .....	14
<b>2 Oracle Solaris</b> での仮想ネットワークの作成および管理 .....	17
ネットワーク仮想化のコンポーネントの構成 .....	17
▼ VNICと etherstub を構成する方法 .....	17
▼ VLAN ID を持つ VNIC を構成する方法 .....	19
仮想ネットワークの構築 .....	20
▼ 仮想ネットワークのためにゾーンを構成する方法 .....	21
▼ ゾーンを VNIC を使用するように再構成する方法 .....	23
▼ プライベート仮想ネットワークを作成する方法 .....	25
VNICのためのその他の管理タスク .....	28
VNICの VLAN ID の変更 .....	29
VNICの MAC アドレスの変更 .....	29
VNICの移行 .....	30
VNIC情報の表示 .....	32
▼ VNICを削除する方法 .....	32

<b>3 Oracle Solaris</b> でのネットワークリソースの管理 .....	35
クライアント、送信リング、および受信リングの操作 .....	35
MACクライアントとリング割り当て .....	35
リング割り当てのためのデータリンクプロパティ .....	37
受信および送信リングを操作するためのコマンド .....	38
リング情報の取得および解釈 .....	38
▼クライアントを構成してリングを割り当てる方法 .....	43
プールおよびCPUの操作 .....	45
▼CPUプールをデータリンク用に構成する方法 .....	47
▼リンクにCPUを割り当てる方法 .....	49
フロー上のリソースの管理 .....	50
▼フローを構成する方法 .....	51
<b>4 Oracle Solaris</b> でのネットワークトラフィックとリソース使用状況の監視 .....	57
ネットワークトラフィックフローの概要 .....	57
トラフィックの統計情報を監視するためのコマンド .....	60
リンク上のネットワークトラフィックに関する統計情報の収集 .....	60
ネットワークデバイス上のネットワークトラフィックの統計情報の取得 .....	61
レーン上のネットワークトラフィックの統計情報の取得 .....	63
リンクアグリゲーション上のネットワークトラフィックの統計情報の取得 .....	64
フロー上のネットワークトラフィックに関する統計情報の収集 .....	65
ネットワークトラフィックのためのネットワークアカウンティングの構成 .....	67
▼ネットワークアカウンティングを設定する方法 .....	67
▼ネットワークトラフィックに関する履歴統計情報を取得する方法 .....	69
索引 .....	73

# はじめに

---

『Oracle Solaris 11.1 での仮想ネットワークの使用』へようこそ。このドキュメントは、Oracle Solaris 11.1 ネットワークの確立に関するシリーズの一部で、Oracle Solaris ネットワークを構成するための基本的なトピックおよび手順について説明しています。このドキュメントの記述は、Oracle Solaris がインストール済みであることが前提です。

## 対象読者

このドキュメントは、Oracle Solaris が動作し、ネットワーク内に構成されているシステムの管理を行うユーザーを対象にしています。このドキュメントを利用するにあたっては、UNIX のシステム管理について少なくとも 2 年の経験が必要です。UNIX システム管理のトレーニングコースに参加することも役に立ちます。

## Oracle サポートへのアクセス

Oracle のお客様は、My Oracle Support を通じて電子的なサポートを利用することができます。詳細は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> を参照してください。聴覚に障害をお持ちの場合は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> を参照してください。

## 表記上の規則

次の表では、このドキュメントで使用される表記上の規則について説明します。

表 P-1 表記上の規則

字体	説明	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 machine_name% you have mail.

表 P-1 表記上の規則 (続き)

字体	説明	例
<b>AaBbCc123</b>	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	<code>machine_name%<b>su</b></code>  <code>Password:</code>
<i>aabbcc123</i>	プレースホルダ: 実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、 <code>rm filename</code> と入力します。
<i>AaBbCc123</i>	書名、新しい単語、および強調する単語を示します。	『ユーザーズガイド』の第6章を参照してください。  キャッシュは、ローカルに格納されるコピーです。  ファイルを保存しないでください。  注: いくつかの強調された項目は、オンラインでは太字で表示されます。

## コマンド例のシェルプロンプト

Oracle Solaris OS に含まれるシェルで使用する、UNIX のデフォルトのシステムプロンプトとスーパーユーザープロンプトを次に示します。コマンド例に示されるデフォルトのシステムプロンプトは、Oracle Solaris のリリースによって異なります。

表 P-2 シェルプロンプト

シェル	プロンプト
Bash シェル、Korn シェル、および Bourne シェル	\$
Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー	#
C シェル	<code>machine_name%</code>
C シェルのスーパーユーザー	<code>machine_name#</code>

# Oracle Solaris でのネットワーク仮想化およびリソース管理

---

この章では、Oracle Solaris でのネットワーク仮想化とリソース制御に関連する基本的な概念について説明します。この章の内容は次のとおりです。

- 7 ページの「ネットワーク仮想化の概要」
- 12 ページの「ネットワークリソースの管理の概要」

これらの機能は、フロー制御を管理したり、システムのパフォーマンスを向上させたり、OS 仮想化、ユーティリティコンピューティング、およびサーバー統合を実現するために必要なネットワーク使用率を構成したりするのに役立ちます。

## ネットワーク仮想化の概要

ネットワーク仮想化とは、ハードウェアのネットワークリソースとソフトウェアのネットワークリソースを1つの管理単位に結合するプロセスのことです。ネットワーク仮想化の目標は、システムとユーザーにネットワークリソースの効率的で、制御され、さらにセキュリティー保護された共有を提供することです。

ネットワーク仮想化の最終生成物は、仮想ネットワークです。仮想ネットワークは、大まかに外部と内部の2つのタイプに分類されます。外部仮想ネットワークは、ソフトウェアによって1つのエンティティーとして管理される複数のローカルネットワークで構成されます。従来の外部仮想ネットワークの構成単位は、スイッチハードウェアと仮想ローカルエリアネットワーク (VLAN) ソフトウェアテクノロジーです。外部仮想ネットワークの例には、大規模な企業ネットワークやデータセンターが含まれます。

このドキュメントでは、内部仮想ネットワークに焦点を置きます。内部仮想ネットワークは、ネットワークインタフェースが少なくとも1枚の物理NIC上に構成されている仮想マシンまたはゾーンを使用した1つのシステムで構成されます。これらのネットワークインタフェースは、仮想ネットワークインタフェースカードまたは

仮想 NIC (vNIC) と呼ばれます。これらのコンテナは、同じローカルネットワーク上に存在するかのように互いに通信することができるため、実質的には単一ホスト上の仮想ネットワークになります。

内部仮想ネットワークの特殊なタイプが、プライベート仮想ネットワークです。プライベート仮想ネットワークは、仮想プライベートネットワーク (VPN) とは異なります。VPN ソフトウェアは、2つのエンドポイントシステム間のセキュリティー保護されたポイントツーポイントリンクを作成します。プライベート仮想ネットワークは、外部システムからはアクセスできないシステム上の仮想ネットワークです。この内部ネットワークのほかの外部システムからの分離は、vNIC を etherstub 上に構成することによって実現されます。etherstub については、次のセクションで説明します。

ネットワークリソースを結合して、内部と外部の両方の仮想ネットワークを構成できます。たとえば、内部仮想ネットワークを含む個々のシステムを、大規模な外部仮想ネットワークの一部である LAN に構成できます。

## ネットワーク仮想化のコンポーネント

Oracle Solaris でのネットワーク仮想化の基本的なコンポーネントを次に示します。

- 仮想ネットワークインタフェースカード (vNIC)
- 仮想スイッチ
- etherstub

vNIC は、物理 NIC と同じデータリンクインタフェースを備えた仮想ネットワークデバイスです。vNIC は、ベースとなるデータリンク上に構成します。構成された vNIC は、物理 NIC のように動作します。さらに、システムのリソースは、vNIC を物理 NIC であるかのように扱います。vNIC には、自動的に生成された MAC アドレスがあります。使用されているネットワークインタフェースによっては、`dladm(1M)` のマニュアルページで説明されているように、vNIC にこのデフォルトのアドレス以外の MAC アドレスを明示的に割り当てることができます。

vNIC をサポートする物理インタフェースの現在の一覧については、[Network Virtualization and Resource Control FAQ \(http://hub.opensolaris.org/bin/view/Project+crossbow/faq\)](http://hub.opensolaris.org/bin/view/Project+crossbow/faq) を参照してください。

vNIC を作成すると、仮想スイッチが自動的に作成されます。Ethernet 設計に従うと、スイッチポートが、そのポートに接続されたホストから送信パケットを受信した場合、そのパケットは同じポート上の宛先に到達できません。仮想ネットワークは同じ NIC を共有するため、この設計は、仮想ネットワークが構成されたシステムでは欠点となります。それらの送信パケットは、スイッチポートを通して外部ネットワークに転送されます。受信パケットは、送信されたのと同じポートを通して戻ることができないため、それらの宛先のゾーンに到達できません。仮想スイッチは、これらのゾーンに、パケットを通過させるための方法を提供します。仮想スイッチは、仮想ネットワークが互いに通信するためのデータパスを開きます。

*etherstub* は擬似 Ethernet NIC です。VNIC は物理リンク上にはではなく、*etherstub* 上に作成できます。*etherstub* 上の VNIC は、システム上の物理 NIC から独立します。*etherstub* を使用すると、システム上のほかの仮想ネットワークと外部ネットワークの両方から切り離されたプライベート仮想ネットワークを作成できます。たとえば、アクセスをネットワーク全体にはではなく、企業の開発者のみに制限したネットワーク環境を作成する場合は、*etherstub* を使用してこのような環境を作成できます。

*etherstub* と VNIC は、Oracle Solaris の仮想化機能の一部にすぎません。通常、これらのコンポーネントは Oracle Solaris のゾーンとともに使用します。VNIC または *etherstub* をゾーンで使用するよう割り当てることによって、1つのシステム内にネットワークを作成できます。ゾーンについては、『[Oracle Solaris 11.1 の管理: Oracle Solaris ゾーン](#)、[Oracle Solaris 10 ゾーン](#)、[およびリソース管理](#)』を参照してください。

これらのコンポーネントを組み合わせ、ゾーンとともに配備することによって、次の図のようなネットワークをシステム内に構築できます。

図 1-1 1つのインタフェースの VNIC 構成

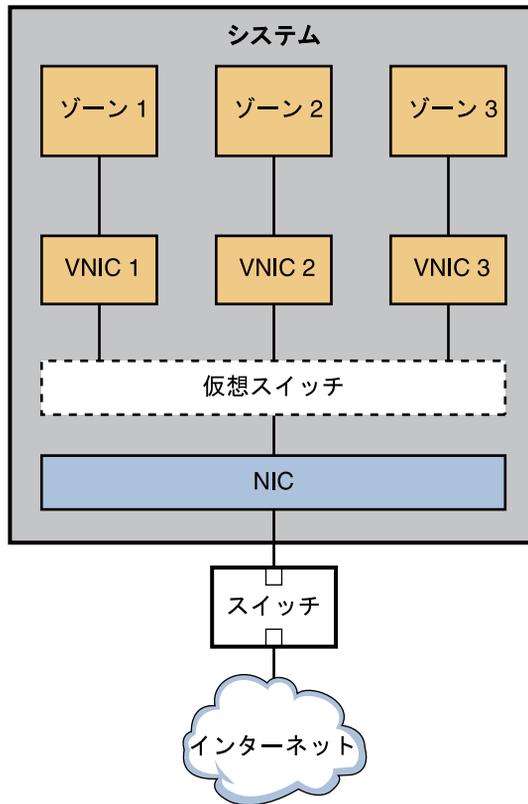


図 1-1 は、1つの NIC を備えた 1つのシステムを示しています。この NIC には 3つの VNIC が構成されています。各 VNIC が 1つのゾーンをサポートします。ゾーン 1、ゾーン 2、およびゾーン 3 が、1つのシステム内の仮想ネットワークを構成しています。これらのゾーンは、それらの各 VNIC を使用して、ほかのゾーンや外部ネットワークと通信します。さらに、これらの 3つの VNIC が、仮想スイッチを通してベースとなる物理 NIC に接続します。仮想スイッチの機能は、そのスイッチのポートに接続されているシステムに外部スイッチによって提供される接続と同等です。

仮想ネットワークが構成されている場合、ゾーンは、仮想ネットワークが存在しないシステムと同じ方法で外部のホストにトラフィックを送信します。トラフィックは、そのゾーンから VNIC を通して仮想スイッチに、さらに物理インタフェースへと転送され、その物理インタフェースによってデータがネットワークに送信されます。

これらのゾーンはまた、システム内で互いにトラフィックを交換することもできます。たとえば、パケットは、ゾーン 1 からその専用の VNIC 1 を通して転送されます。

す。そのトラフィックは次に、仮想スイッチを通して VNIC3 に転送されます。次に、VNIC3 がそのトラフィックをゾーン3 に通します。このトラフィックがシステムを離れることはないため、Ethernet の制限には違反しません。

あるいは、etherstub に基づいて仮想ネットワークを作成できます。etherstub は完全にソフトウェアベースであるため、仮想ネットワークの基礎としてのネットワークインタフェースを必要としません。

## 仮想ネットワークを実装すべきユーザー

Oracle の Sun サーバー上のリソースを統合する必要がある場合は、VNIC と仮想ネットワークの実装を検討してください。ISP、電気通信会社、および大規模な金融機関の統合担当者は、次のネットワーク仮想化機能を使用して、所有するサーバーとネットワークのパフォーマンスを向上させることができます。

- NIC ハードウェア (ハードウェアリングをサポートする強力な、新しいインタフェースを含む)
- VNIC のための複数の MAC アドレス
- より新しいインタフェースによって提供される大量の帯域幅

分離、セキュリティー、および柔軟性を大きく損なうことなく、多数のシステムを、複数のゾーンまたは仮想マシンを含む 1 つのシステムに置き換えることができます。

ネットワーク仮想化の利点のデモについては、[Consolidating the Data Center With Network Virtualization](http://download.oracle.com/otndocs/tech/OTN_Demos/data-center-consolidation.html) ([http://download.oracle.com/otndocs/tech/OTN\\_Demos/data-center-consolidation.html](http://download.oracle.com/otndocs/tech/OTN_Demos/data-center-consolidation.html)) を参照してください。

## 仮想化コンポーネントを構成するためのコマンド

VNIC を作成するには、`dladm create-vnic` コマンドを使用します。

```
# dladm create-vnic -l link [-v vid] vnic
```

`-l link` VNIC が構成されているデータリンクの名前を示します。

`-v vid` VNIC を VLAN として作成する場合は、その VNIC の VLAN ID を示します。このオプションは必須ではありません。VLAN ID を持つ VNIC を構成するには、[19 ページの「VLAN ID を持つ VNIC を構成する方法」](#) を参照してください。VLAN についての詳細は、『Oracle Solaris 11.1 ネットワークパフォーマンスの管理』の第 3 章「VLAN の操作」を参照してください。

`vnic` VNIC の名前を示します。

---

注- MAC アドレスや、VNIC に関連付ける CPU などの、VNIC のその他のプロパティを構成できます。これらのプロパティの一覧については、[dladm\(1M\)](#) のマニュアルページを参照してください。特定のプロパティ変更は、VNIC に関してのみ機能します。たとえば、`dladm create-vnic` コマンドを使用すると、VNIC を VLAN として作成するために MAC アドレスを構成したり、VLAN ID を割り当てたりすることができます。ただし、`dladm create-vlan` コマンドを使用して、直接 VLAN の MAC アドレスを構成することはできません。

---

VNIC はデータリンク上に一度に 1 つのみ作成できます。VNIC には、データリンクとして、必要に応じてさらに構成できるリンクプロパティがあります。[12 ページ](#)の「[リソース制御のためのデータリンクプロパティ](#)」には、システム内のネットワークリソースの使用を管理するためのこれらのプロパティの一部が示されています。

`etherstub` を作成するには、`dladm create-ether` コマンドを使用します。

```
# dladm create-ether etherstub
```

VNIC または `etherstub` の作成は、仮想ネットワークの構成における予備的な手順にすぎません。これらのコンポーネントを使用してシステム上に仮想ネットワークを作成するには、[第 2 章「Oracle Solaris での仮想ネットワークの作成および管理」](#)を参照してください。

## ネットワークリソースの管理の概要

このセクションでは、システム上のネットワークリソースの使用を管理するために使用できる各種の方法について説明します。

### リソース制御のためのデータリンクプロパティ

Oracle Solaris 11 では、ネットワークリソースを管理することによって、サービスの品質 (QoS) がより容易に、かつ動的に得られます。ネットワークリソースの管理では、ネットワークリソースに関連するデータリンクプロパティを設定します。これらのプロパティを設定することによって、特定のリソースのどれだけの量をネットワークプロセスに使用できるかを判断します。たとえば、リンクを、ネットワークプロセスのために排他的に予約された特定の数の CPU に関連付けることができます。または、リンクに、特定のタイプのネットワークトラフィックを処理するための特定の帯域幅を割り当てることができます。

リソースプロパティが定義されると、その新しい値はただちに有効になります。この方法によって、リソースの管理が柔軟になります。リソースのプロパティは、リンクを作成するときに設定できます。あるいは、たとえば一定期間にわたってリソース使用状況を調査して、リソースをより適切に割り当てる方法を特定したあとなど、あとからこれらのプロパティを設定することもできます。リソースを割り当てるための手順は、仮想ネットワーク環境と従来の物理ネットワークの両方に適用されます。たとえば、ネットワークリソースに関連したプロパティを設定するには、`dladm set-linkprop` コマンドを使用します。この同じ構文が、物理データリンクと仮想データリンクの両方に対して使用されます。

ネットワークリソースの管理は、トラフィックのための専用のレーンを作成することに相当します。特定のタイプのネットワークパケットの要求にこたえるためにさまざまなリソースを結合すると、これらのリソースによって、このパケットのためのネットワークレーンが形成されます。ネットワークレーンごとに、リソースを異なる方法で割り当てることができます。たとえば、ネットワークトラフィックの負荷がもっとも高いレーンに、より多くのリソースを割り当てることができます。ネットワークレーンを実際のニーズに応じて分散されるように構成することによって、パケットを処理するシステムの効率が向上します。ネットワークレーンについての詳細は、57 ページの「[ネットワークトラフィックフローの概要](#)」を参照してください。

ネットワークリソースの管理は、次のタスクに役立ちます。

- ネットワークのプロビジョニング
- サービスレベル契約の確立
- クライアントへの請求
- セキュリティーに関する問題の診断

複雑な QoS 規則の定義を必要とすることなく、個々のシステム上のデータトラフィックの分離、優先順位付け、追跡、制御などを行うことができます。

## フローを使用することによるネットワークリソースの管理

フローは、パケットを分類することにより、これらのパケットの処理でのリソースの使用方法をさらに制御するためのカスタマイズされた方法です。ネットワークパケットは、属性に従って分類できます。属性を共有するパケットによってフローが構成され、これらのパケットには特定のフロー名が付けられます。次に、このフローに特定のリソースを割り当てることができます。

フローを作成するための基盤として機能する属性は、パケットのヘッダー内の情報から導き出されます。パケットトラフィックは、次のいずれかの属性に従ってフローに構成できます。

- IPアドレス
- トランスポートプロトコル名 (UDP、TCP、または SCTP)
- アプリケーションのポート番号 (たとえば、FTP ではポート 21)
- DS フィールドの属性 (IPv6 パケットでの QoS にのみ使用されます) DS フィールドについての詳細は、『[Oracle Solaris 11.1 での IP サービス品質の管理](#)』を参照してください。

フローは、この一覧内の 1 つの属性にのみ基づいて作成できます。たとえば、使用されているポート (FTP でのポート 21 など) か、または IP アドレス (特定の発信元 IP アドレスからのパケットなど) に従ってフローを作成できます。ただし、ポート番号 21 上で受信された、指定された IP アドレスからのパケットのためのフローを作成することはできません。同様に、IP アドレス 192.168.1.10 からのすべてのトラフィックのためのフローを作成したあと、192.168.1.10 上のトランスポートレイヤートラフィックのためのフローを作成することはできません。そのため、システム上に、各フローが異なる属性に基づいた複数のフローを構成できます。

## ネットワークリソースの管理のためのコマンド

ネットワークリソースを割り当てるために使用するコマンドは、データリンクまたはフローのどちらに対して直接作業しているかによって異なります。

- データリンクの場合は、リンクの作成中にプロパティを設定しているのか、または既存のリンクのプロパティを設定しているのかに応じて、該当する `dladm` サブコマンドを使用します。リンクの作成と、そのリンクへのリソースの割り当てを同時に行うには、次の構文を使用します。

```
# dladm create-vnic -l link -p property=value[,property=value] vnic
```

ここで、`link` には物理リンクまたは仮想リンクを指定できます。

既存のリンクのプロパティを設定するには、次の構文を使用します。

```
# dladm set-linkprop -p property=value[,property=value] link
```

リソース割り当てのために設定できるリンクプロパティを次に示します。

- 帯域幅 - 特定のリンクの使用のためのハードウェアの帯域幅を制限できます。
- NIC リング - NIC がリング割り当てをサポートしている場合は、その送信および受信リングを、データリンクによる使用に専用に割り当てることができます。NIC リングについては、[35 ページの「クライアント、送信リング、および受信リングの操作」](#)で説明されています。

- **CPU プール** – CPU のプールは一般に、作成されて特定のゾーンに関連付けられます。これらのプールをデータリンクに割り当てることにより、関連付けられたゾーンのネットワークプロセスを管理するために一連の CPU を予約できます。CPU およびプールについては、[45 ページ](#)の「**プールおよび CPU の操作**」で説明されています。
- **CPU** – 複数の CPU を備えたシステムでは、指定された数の CPU を特定のネットワーク処理に専用に割り当てるができます。
- フローの場合は、`flowadm` サブコマンドを使用します。フロー上のリソースの管理は、データリンク上のリソースを管理するための方法に対応しています。フローの作成と、そのフローへのリソースの追加を同時に行うには、次の構文を使用します。

```
# flowadm add-flow -l link -a attribute=value[,attribute=value] \
-p property=value[,property=value] flow
```

フローを特徴づける一連の定義された属性によって、システムのフロー制御ポリシーが構成されます。

既存のフローのプロパティを設定するには、次の構文を使用します。

```
# flowadm set-flowprop -p property=value[,property=value] flow
```

フローに割り当てることのできるリソース割り当てのためのプロパティは、リンクに直接割り当てられるプロパティと同じです。ただし、現在、フローに関連付けることができるのは帯域幅プロパティだけです。プロパティを設定するためのコマンドはデータリンクの場合とフローの場合で異なりますが、構文は似ています。帯域幅プロパティを構成するには、[51 ページ](#)の「**フローを構成する方法**」にある例を参照してください。

`flowadm` コマンドについての詳細は、[flowadm\(1M\)](#) のマニュアルページを参照してください。 `flowadm` コマンドで使用するサブコマンドの一覧を表示するには、次のように入力します。

```
# flowadm help
The following subcommands are supported:
Flow      : add-flow      remove-flow      reset-flowprop
            set-flowprop  show-flow        show-flowprop
For more info, run: flowadm help <subcommand>.
```



## Oracle Solaris での仮想ネットワークの作成および管理

---

この章には、1つのシステム内に仮想ネットワークを構成するためのタスクが含まれています。

この章の内容は次のとおりです。

- 17 ページの「ネットワーク仮想化のコンポーネントの構成」
- 20 ページの「仮想ネットワークの構築」
- 28 ページの「VNIC のためのその他の管理タスク」

仮想ネットワークの概要については、第1章「Oracle Solaris でのネットワーク仮想化およびリソース管理」を参照してください。

### ネットワーク仮想化のコンポーネントの構成

Oracle Solaris 11 では、etherstub と VNIC がネットワーク仮想化の基本的なコンポーネントです。このセクションでは、仮想ネットワークを構築する準備として、これらのコンポーネントを構成する手順について説明します。これらのコンポーネントについては、8 ページの「ネットワーク仮想化のコンポーネント」を参照してください。

次の手順について説明します。

- 17 ページの「VNIC と etherstub を構成する方法」
- 19 ページの「VLAN ID を持つ VNIC を構成する方法」

#### ▼ VNIC と etherstub を構成する方法

VNIC は、仮想ネットワークを外部ネットワークに接続します。VNIC ではまた、VNIC とともに自動的に作成される仮想スイッチを通してゾーンが互いに通信することもできます。仮想ネットワークがゾーン間の内部的なトラフィックや、外部

の LAN およびインターネットとのトラフィックをホストするには、各ゾーンに独自のインタフェースが存在する必要があります。そのため、この手順を仮想ネットワークに属するゾーンの数だけ繰り返す必要があります。

1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 (オプション) etherstub を作成します。

```
# dladm create-etherstub etherstub
```

この手順は、外部システムからはアクセスできないように制限する、プライベート仮想ネットワークを作成する場合にのみ実行します。プライベート仮想ネットワークについては、7 ページの「ネットワーク仮想化の概要」を参照してください。

データリンクと同様に、etherstub にも、ネットワークの設定にとって意味のある任意の方法で名前を付けることができます。カスタマイズされた名前の作成に関するガイドラインについては、『Oracle Solaris 11 ネットワーキングの紹介』の「有効なリンク名のための規則」を参照してください。

3 VNIC を作成します。

```
# dladm create-vnic -l datalink [-v vid] vnic
```

VNIC をプライベート仮想ネットワークのために作成している場合は、datalink に etherstub を指定します。コマンド構文に -v vid を含めるのは、VNIC を VLAN として作成している場合だけです。ここで、vid はその VNIC の VLAN ID を示します。それ以外の場合は、このオプションを省略します。

VNIC を VLAN として作成している場合は、VLAN としての VNIC に固有の追加の手順について 19 ページの「VLAN ID を持つ VNIC を構成する方法」を参照してください。

VNIC に任意の名前を割り当てることができます。VNIC にカスタマイズされた名前を割り当てるには、『Oracle Solaris 11 ネットワーキングの紹介』の「有効なリンク名のための規則」を参照してください。

4 VNIC 上に IP インタフェースを作成します。

```
# ipadm create-ip interface
```

5 VNIC インタフェースに静的 IP アドレスを割り当てます。

```
# ipadm create-addr -a address interface
```

-a address IP アドレスを指定します。これは CIDR 表記にすることができます。

interface 前の手順で作成した VNIC を指定します。

静的 IP アドレスは、IPv4 アドレスと IPv6 アドレスのどちらでもかまいません。IP アドレスの構成についての詳細は、『Oracle Solaris 11.1 での固定ネットワーク構成を使用したシステムの接続』の「IP インタフェースを構成する方法」を参照してください。

IP アドレスの構成についての詳細は、『Oracle Solaris 11.1 での固定ネットワーク構成を使用したシステムの接続』の「IP インタフェースを構成する方法」を参照してください。

- 6 /etc/hosts ファイルにアドレス情報を追加します。

## ▼ VLAN ID を持つ VNIC を構成する方法

仮想ネットワークでは、VLAN トラフィックをホストするために VLAN ID を持つ VNIC を構成できます。また、個々の VNIC の VLAN 構成をネットワークに伝播するために、リンクプロパティ `vlan-announce` を設定することもできます。

通常の VLAN リンクとは異なり、VLAN として構成された VNIC には独自の MAC アドレスがあります。VNIC 以外の VLAN については、『Oracle Solaris 11.1 ネットワークパフォーマンスの管理』の第 3 章「VLAN の操作」を参照してください。

---

注-次の手順には、VLAN ID を持つ VNIC を作成し、その VNIC が VLAN トラフィックを処理できるようにするための適切なプロパティを設定する手順のみが含まれています。このプロパティを有効にすると中間のポートやスイッチは自動的に更新されますが、エンドポイントは、これらのポイントで VLAN を定義するように個別に構成する必要があります。

---

- 1 管理者になります。  
詳細は、『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 VLAN ID を持つ VNIC を作成します。

```
# dladm create-vnic -l link -v vid vnic
```

- 3 VNIC の VLAN 構成をネットワークにブロードキャストします。

```
# dladm set-linkprop -p vlan-announce=gvrp link
```

この手順では、接続されているスイッチに VLAN ID を自動的に登録する GARP VLAN 登録プロトコル (GVRP) クライアントシステムを有効にします。デフォルトでは、`vlan-announce` プロパティが `off` に設定されているため、ネットワークに VLAN ブロードキャストメッセージは送信されません。このプロパティを `gvrp` に設定すると、ネットワークデバイスの自動 VLAN ポート構成を有効にするため

に、そのリンクの VLAN 構成が伝播されます。これにより、これらのデバイスで VLAN トラフィックを受け入れ、転送することができます。

- 4 (オプション)VLAN ブロードキャスト間の待機時間を構成するには、`gvrp-timeout` プロパティを設定します。

```
# dladm set-linkprop -p gvrp-timeout=time link
```

ここで、`time` はミリ秒単位です。デフォルト値は 250 ミリ秒です。負荷の高いシステムでは、VLAN 情報を再ブロードキャストするときに、より短い間隔が必要になることがあります。このプロパティを使用すると、この間隔を調整できます。

- 5 (オプション)プロパティ `vlan-announce` および `gvrp-timeout` の値を表示するには、次のコマンドを使用します。

```
# dladm show-linkprop -p vlan-announce,gvrp-timeout
```

### 例 2-1 VLAN としての VNIC の構成

この例では、VLAN ID を持つ VNIC を作成したあと、その VLAN 構成をネットワークに通知できるようにします。

```
# dladm create-vnic -l net0 -v 123 vnic0
# dladm set-linkprop -p vlan-announce=gvrp net0
# dladm show-linkprop -p vlan-announce,gvrp-timeout net0
LINK  PROPERTY      PERM  VALUE  DEFAULT  POSSIBLE
net0  vlan-announce  rw    gvrp   off      gvrp,off
net0  gvrp-timeout   rw    250    250     --
```

## 仮想ネットワークの構築

仮想ネットワークによって、ゾーンと仮想化のコンポーネントが結合されます。ゾーンは必要な数だけ、かつシステムがサポートできる数だけ作成します。各ゾーンには、独自の仮想インタフェースがあります。システム内の各ゾーンは互いに通信できます。仮想ネットワークは、全体として、より大きな外部ネットワーク上の宛先に接続します。

仮想ネットワークの構築は、`etherstub` または VNIC を構成する 1 つ以上の手順、およびゾーンを構成する手順で構成されています。これらの一連の手順は独立していますが、仮想ネットワークの構成を完了するには、この両方を実行する必要があります。

このセクションの手順は、次の前提に基づいて進められます。

- システム上の仮想ネットワークは、3 つのゾーンで構成されています。これらのゾーンは、構成の中の異なる段階にあります。最初のゾーンは新しいゾーンとして作成されます。2 番目のゾーンはシステム上にすでに存在し、VNIC を使用する

ために再構成する必要があります。3番目のゾーンは、プライベート仮想ネットワークとして指定されます。そのため、これらの手順では、仮想ネットワークのためにゾーンを準備するためのさまざまな方法が示されます。

- システムの物理インターフェースには、IPアドレス **192.168.3.70** が構成されます。
- ルーターのIPアドレスは **192.168.3.25** です。

このセクションの各手順では、手順により具体的なコンテキストを提供するために、より詳細な内容がシナリオに追加されます。

仮想ネットワークを構築する場合、一部の手順は大域ゾーンで実行され、一部の手順は非大域ゾーンで実行されます。わかりやすくするために、各手順のあとの例にあるプロンプトは、特定のコマンドがどのゾーンで発行されているかを示しています。ただし、プロンプトに表示される実際のパスは、使用しているシステムに指定されたプロンプトに応じて異なることがあります。

このセクションでは、次の手順について説明します。

- [21 ページの「仮想ネットワークのためにゾーンを構成する方法」](#)
- [23 ページの「ゾーンを VNIC を使用するように再構成する方法」](#)
- [25 ページの「プライベート仮想ネットワークを作成する方法」](#)

## ▼ 仮想ネットワークのためにゾーンを構成する方法

この手順では、新しい VNIC を備えた新しいゾーンを構成する方法について説明します。この手順には、ネットワーク仮想化に関連した手順のみが含まれていることに注意してください。ゾーンの構成に関するより詳細な手順については、『[Oracle Solaris 11.1 の管理: Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理](#)』の第 17 章「[非大域ゾーンの計画と構成\(タスク\)](#)」を参照してください。

この手順では、仮想ネットワークのためのこの最初のゾーンが新しいゾーンとして作成されることを前提にしています。

### 1 管理者になります。

詳細は、『[Oracle Solaris 11.1 の管理: セキュリティサービス](#)』の「[割り当てられている管理権限を使用する方法](#)」を参照してください。

### 2 VNIC を構成します。

[17 ページの「VNIC と etherstub を構成する方法」](#)を参照してください。ただし、この特定の手順では、etherstub を作成する手順を省略します。

### 3 ゾーンを作成します。

```
global# zonecfg -z zone
```

ゾーンを作成する場合は、`ip-type` パラメータを `exclusive` に設定し、今作成した VNIC をゾーンの物理インタフェースとして割り当てるようにしてください。

- 4 ゾーン構成モードを終了するために、構成を確認してからコミットします。

- 5 ゾーンをインストールします。

```
global# zoneadm -z zone install
```

---

注-インストール処理には時間がかかることがあります。

---

- 6 ゾーンを起動します。

```
global# zoneadm -z zone boot
```

- 7 ゾーンが完全にブートしたら、そのゾーンにログインします。

```
# zlogin -C zone
```

- 8 入力を求められたら、情報を指定します。

ほとんどの情報は、選択リストから選択することによって指定されます。通常は、デフォルトオプションで十分です。仮想ネットワークを構成するには、次の情報を指定または確認する必要があります。

- ゾーンのホスト名(たとえば、zone1)。
- ゾーンの VNIC の IP アドレスに基づいたゾーンの IP アドレス。
- IPv6 を有効にするかどうか。
- 仮想ネットワークを含むシステムがサブネットの一部であるかどうか。
- IP アドレスのネットマスク。
- デフォルトルート。これには、仮想ネットワークが構築されている物理インタフェースの IP アドレスを指定できます。

必要な情報を指定すると、ゾーンが再起動します。

## 例 2-2 仮想ネットワークのためのゾーンの構成

この例には、zone1 を作成するための詳細な手順が含まれています。ただし、仮想ネットワークの作成に関連するゾーンパラメータのみが一覧表示されています。

```
global # zonecfg -z zone1
zonecfg:zone1> create
zonecfg:zone1> set zonepath=/export/home/zone1
zonecfg:zone1> set autoboot=true
zonecfg:zone1> set ip-type=exclusive
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic1
```

```

zonecfg:zone1:net> end
zonecfg:zone1> verify
zonecfg:zone1> commit
zonecfg:zone1> exit

global# zoneadm -z zone1 install
Preparing to install zone <zone1>
Creating list of files to copy from the global zone.
.
.
Zone <zone1> is initialized.

global# zoneadm -z zone1 boot

zlogin -C zone1
What type of terminal are you using?
.
.
8) Sun Workstation
9) Televideo 910
10) Televideo 925
11) Wyse Model 50
12) X Terminal Emulator (xterms)
13) CDE Terminal Emulator (dtterm)
14) Other
Type the number of your choice and press Return: 13
.
(More prompts)
..

```

ネットワーク情報については、次の情報が指定されます。

```

Hostname: zone1
IP address: 192.168.3.80
System part of a subnet: Yes
Netmask: 255.255.255.0
Enable IPv6: No
Default route: 192.168.3.70
Router IP address: 192.168.3.25

```

## ▼ ゾーンを VNIC を使用するように再構成する方法

この手順では、仮想ネットワーク内の 2 番目のゾーンを参照します。そのゾーンはすでに存在しますが、その現在の構成のために、仮想ネットワークの一部になることができません。具体的には、ゾーンの IP タイプが共有タイプであり、その現在のインタフェースは net0 です。これらの構成の両方を変更する必要があります。

### 1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

## 2 VNICを作成します。

```
global# dladm create-vnic [-v vid] -l datalink vnic
```

ここで、*vid*は、VNICに割り当てるVLAN IDを示します。このVLAN IDは、VNICをVLANとして作成する場合にのみ指定します。

VNICのインタフェースはまだ構成しないでください。この手順は、一連の手順の後半で実行します。

## 3 ゾーンのIPタイプを共有から排他的に変更します。

```
global# zonecfg -z zone
zonecfg:zone1> set ip-type=exclusive
zonecfg:zone1>
```

## 4 ゾーンのインタフェースをVNICを使用するように変更します。

```
zonecfg:zone1> remove net physical=NIC
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic
zonecfg:zone1:net> end
zonecfg:zone1>
```

## 5 実装した変更を確認してコミットしたあと、そのゾーンを終了します。

```
zonecfg:zone1 verify
zonecfg:zone1> commit
zonecfg:zone1> exit
global#
```

## 6 ゾーンをリブートします。

```
global# zoneadm -z zone reboot
```

## 7 ゾーンにログインします。

```
global# zlogin zone
```

## 8 VNICに有効なIPアドレスを構成します。

VNICに静的アドレスを割り当てる場合は、次を入力します。

```
zone# ipadm create-addr -a address interface
```

ここで、*address*にはCIDR表記を使用できます。

## 9 大域ゾーンから、*/etc/hosts* ファイルにアドレス情報を追加します。

### 例 2-3 VNICを使用するためのゾーン構成の再構成

この例では、*zone2*がすでに共有ゾーンとして存在します。このゾーンはまた、仮想リンクではなく、システムのプライマリインタフェースを使用しています。*vnic2*を使用するように*zone2*を変更する必要があります。*vnic2*を使用するには、まず

zone2 の IP タイプを排他的に変更する必要があります。仮想ネットワークに関連する情報に注目しているために、一部の出力が切り捨てられていることに注意してください。

```
global# dladm create-vnic -l net0 vnic2

global# zonecfg -z zone2
zonecfg:zone1> set ip-type=exclusive
zonecfg:zone1> remove net physical=net0
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic2
zonecfg:zone1:net> end
zonecfg:zone1> verify
zonecfg:zone1> commit
zonecfg:zone1> exit
global# zoneadm -z zone2 reboot

global# zlogin zone2
zone2# ipadm create-ip vnic2
zone2# ipadm create-addr -a 192.168.3.85/24 vnic2
ipadm: vnic2/v4

zone2# exit

global# vi /etc/hosts
#
::1          localhost
127.0.0.1    localhost
192.168.3.70 loghost    #For net0
192.168.3.80 zone1      #using vnic1
192.168.3.85 zone2      #using vnic2
```

## ▼ プライベート仮想ネットワークを作成する方法

次の手順は、仮想ネットワークの3番目のゾーンを構成する方法を説明しています。このゾーンは仮想ネットワークの一部ですが、外部システムからはアクセスできません。切り離されたゾーンがシステムを超えてネットワークトラフィックを送信できるようにするには、ネットワークアドレス変換 (NAT) を使用する必要があります。NAT は、VNIC のプライベート IP アドレスを物理ネットワークインタフェースのルーティング可能な IP アドレスに変換します。ただし、プライベート IP アドレス自体は外部ネットワークから見えません。NAT についての詳細は、『[Oracle Solaris 11.1 でのネットワークのセキュリティ保護](#)』の「IP フィルタの NAT 機能の使用」を参照してください。

etherstub の使用によって、通常の仮想ネットワークとプライベート仮想ネットワークの主な違いが生まれます。プライベート仮想ネットワークでは、ゾーンに割り当てられた VNIC は etherstub 上に構成されます。そのため、システムを通過するネットワークトラフィックから切り離されます。

この手順では、ゾーンはすでに存在しているものの、現在はどのインタフェースにも関連付けられていないことを前提にしています。

- 1 管理者になります。

詳細は、『Oracle Solaris 11.1の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 **etherstub** を作成します。

```
global# dladm create-etherstub etherstub
```

- 3 **etherstub** 上に **VNIC** を作成します。

```
global# dladm create-vnic -l etherstub vnic
```

VNICのインタフェースはまだ構成しないでください。この手順は、一連の手順の後半で実行します。

- 4 ゾーンに **VNIC** を割り当てます。

```
global# zonecfg -z zone
zone# set physical=vnic
```

- 5 実装した変更を確認してコミットしたあと、そのゾーンを終了します。

```
zonecfg:zone1 verify
zonecfg:zone1> commit
zonecfg:zone1> exit
global#
```

- 6 ゾーンにログインします。

```
# zlogin zone
```

- 7 ゾーンで、現在そのゾーンに割り当てられている **VNIC** 上に **IP** インタフェースを作成します。

```
# ipadm create-ip interface
```

- 8 **VNIC** に有効な **IP** アドレスを構成します。

VNICに静的アドレスを割り当てる場合は、次を入力します。

```
zone# ipadm create-addr -a address interface
```

ここで、*address* にはCIDR表記を使用できます。

- 9 大域ゾーンから、**/etc/hosts** ファイルにアドレス情報を追加します。

- 10 大域ゾーンから、**IP** 転送を実行するようにプライマリインタフェースを設定します。

```
# ipadm set-ifprop -p forwarding=on -m ipv4 primary-interface
```

注 - Oracle Solaris 11 では通常、プライマリインタフェースは `net0` の名前を使用します。

- 11 大域ゾーンから、プライマリインタフェースの `/etc/ipnat.conf` ファイル内にネットワークアドレス変換 (NAT) を構成します。
- 12 IP フィルタサービスを起動して NAT を有効にします。  
# `svcadm enable network/ipfilter`
- 13 ゾーンをリブートします。  
# `zoneadm -z zone reboot`

## 例 2-4 プライベート仮想ネットワーク構成の作成

この例では、`zone3` を、プライベートネットワークとして切り離されるように構成します。また、仮想プライベートネットワークがホストの外部にパケットを送信できるようにする一方、そのプライベートアドレスを引き続き外部ネットワークから見えないようにするために、NAT と IP 転送も構成します。このゾーンには、排他的な IP タイプがすでに構成されています。ただし、IP インタフェースは割り当てられていません。

```
global# dladm create-etherstub ether0
global# dladm create-vnic -l ether0 vnic3
global# zonecfg -z zone3
zonecfg:zone3> add net
zonecfg:zone3:net> set physical=vnic3
zonecfg:zone3:net> end
zonecfg:zone3> verify
zonecfg:zone3> commit
zonecfg:zone3> exit
global#

global# zlogin zone3
zone3# ipadm create-ip vnic3
zone3# ipadm create-addr -a 192.168.0.10/24 vnic3
ipadm: vnic3/v4
zone3# exit

global# cat /etc/hosts
::1          localhost
127.0.0.1    localhost
192.168.3.70 loghost     #For net0
192.168.3.80 zone1      #using vnic1
192.168.3.85 zone2      #using vnic2
192.168.0.10 zone3      #using vnic3

global# ipadm set-ifprop -p forwarding=on -m ipv4 vnic3

global# vi /etc/ipf/ipnat.conf
```

```
map vnic3 192.168.0.0/24 -> 0/32 portmap tcp/udp auto
map vnic3 192.168.0.0/24 -> 0/32
```

```
global# svcadm enable network/ipfilter
global# zoneadm -z zone3 boot
```

## VNICのためのその他の管理タスク

このセクションでは、基本構成を実行したあとにVNIC上で実行できるタスクについて説明します。このセクションの内容は次のとおりです。

- 29 ページの「VNIC の VLAN ID の変更」
- 29 ページの「VNIC の MAC アドレスの変更」
- 30 ページの「VNIC の移行」
- 32 ページの「VNIC 情報の表示」
- 32 ページの「VNIC を削除する方法」

---

注 - VNIC を VLAN として構成できます。並列サブコマンド `dladm modify-vlan` を使用すると、`dladm create-vlan` コマンドを使用して作成された直接 VLAN を変更できません。VLAN または VNIC として構成された VLAN のどちらを変更しているかに応じて、適切なサブコマンドを使用する必要があります。`dladm show-vlan` サブコマンドで表示される VLAN に対しては、`modify-vlan` サブコマンドを使用します。`dladm show-vnic` サブコマンドで表示される VNIC (VLAN ID を持つ VNIC を含む) に対しては、`modify-vnic` サブコマンドを使用します。直接 VLAN を変更するには、『[Oracle Solaris 11.1 ネットワークパフォーマンスの管理](#)』の「[VLAN の変更](#)」を参照してください。

---

次の2つのタイプのVNIC変更が使用可能です。

- グローバルな変更では、特定のデータリンク上のすべてのVNICの属性が一度に変更されます。VNICを変更するベースとなるデータリンクを識別するには、`-L` オプションを使用します。
- 選択的な変更では、選択されたVNICの属性が変更されます。`-L` オプションを使用してベースとなるデータリンクを識別する代わりに、属性を変更するVNICを指定します。

VLAN ID、MAC アドレス、ベースとなるリンクの各属性を変更できます。ベースとなるリンクを変更すると、VNICが別のデータリンクに移動されます。以降のセクションでは、これらの変更について詳細に説明します。

## VNICのVLAN IDの変更

VNICのVLAN IDを変更するには、次のコマンドのいずれかを使用します。

- `dladm modify-vnic -v vid -L datalink`

このコマンドでは、*vid*が、VNICに割り当てる新しいVLAN IDを指定します。*datalink*は、VNICが構成されているベースとなるリンクを示します。このコマンド構文を使用できるのは、データリンク上に1つのVNICが存在する場合だけです。複数のVNICが構成されたデータリンク上では、これらのVNICに一意のVLAN IDを割り当てる必要があるため、このコマンドは失敗します。

- `dladm modify-vnic -v vid vnic`

このコマンドは、1つのデータリンク上の複数のVNICの一意のVLAN IDを変更するために使用します。各VLAN IDは同じデータリンク上のVNICで一意であるため、VLAN IDは一度に1つずつ変更する必要があります。*net0*上に構成された *vnic0*、*vnicb0*、および *vnicc0* のVLAN IDを変更するとします。次のように進めます。

```
# dladm modify-vnic -v 123 vnic0
# dladm modify-vnic -v 456 vnicb0
# dladm modify-vnic -v 789 vnicc0
```

- `dladm modify-vnic -v vid vnic, vnic, [...]`

このコマンドは、各VNICが異なるデータリンク上に存在する場合に、VNICのVLAN IDをグループとして変更するために使用します。*vnic0*、*vnic1*、および *vnic2* のVLAN IDを変更するとします。これらのVNICはそれぞれ、*net0*、*net1*、および *net2* 上に構成されています。次のコマンドを使用します。

```
# dladm modify-vnic -v 123 vnic0,vnic1,vnic2
```

## VNICのMACアドレスの変更

VNICには、一意のMACアドレスがあります。これらのアドレスを変更するには、特定の環境に応じて、次のコマンドのいずれかを使用します。

- `dladm modify-vnic -m mac-address vnic`

このコマンドは、特定のVNICに特定のMACアドレスを割り当てるために使用します。

- `dladm modify-vnic -m random -L datalink`

このコマンドは、データリンク上のすべてのVNICのMACアドレスを変更するグローバルな変更を実行します。システムは、各VNICに一意のMACアドレスを自動的に割り当てます。このコマンドでは、`-m random` オプションは `-m auto` オプションと同等です。

- `dladm modif-vnic -m random vnic, vnic,[...]`

このコマンドは、VNICの選択的な変更を実行します。グローバルな変更と選択的な変更のどちらも、`-m` オプションに `random` を指定することに注意してください。

VNICのVLAN IDとMACアドレスを1つのコマンドで変更できます。ただし、複数のVNIC属性をグローバルに変更するコマンドの使用は、予期しない動作を引き起こす可能性があるため注意してください。VNICのグループの複数の属性をすべて同時に変更するよりも、1つのVNICごとに複数の属性を変更することをお勧めします。

次の例は、VNICのVLAN IDとMACアドレスを変更する前とあとの出力を示しています。

```
# dladm show-vnic vnic0
LINK      OVER      SPEED    MACADDRESS      MACADDRTYPE     VID
vnic0     net0      1000    2:8:20:ec:c4:1d random           0
# dladm modify-vnic -m random -v 123 vnic0
# dladm show-vnic vnic0
LINK      OVER      SPEED    MACADDRESS      MACADDRTYPE     VID
vnic0     net0      1000    2:8:20:0:1:2   random           123
```

## VNICの移行

VNICを削除して再構成することなく、1つ以上のVNICを、あるベースとなるデータリンクから別のベースとなるデータリンクに移動できます。ベースとなるリンクには、物理リンク、リンクアグリゲーション、または `etherstub` を指定できます。

VNICを正常に移行するには、VNICの移動先のベースとなるデータリンクが、そのVNICのデータリンクプロパティに対応している必要があります。これらのプロパティがサポートされていない場合は、移行が失敗し、ユーザーに通知されます。移行が成功したあと、そのVNICがネットワークに接続されたままであれば、VNICを使用しているすべてのアプリケーションが引き続き正常に動作します。

VNICの移行のあと、データリンクの状態、リンク速度、MTUサイズなどの、ハードウェアに依存した特定のプロパティは変更される可能性があります。これらのプロパティの値は、VNICの移行先のデータリンクから継承されます。

VNICはまた、グローバルに移行することも、選択的に移行することもできます。グローバルな移行では、あるデータリンク上のすべてのVNICを別のデータリンクに移行します。グローバルな移行を実行するには、ソースデータリンクとターゲットデータリンクを指定するだけで済みます。次の例では、すべてのVNICを `ether0` から `net1` に移動します。

```
# dladm modify-vnic -l net1 -L ether0
```

ここで

- `-l datalink` は、VNIC の移行先の宛先データリンクを示します。
- `-L datalink` は、VNIC が構成されている元のデータリンクを示します。

---

注-ソースデータリンクの前に宛先データリンクを指定する必要があります。

---

VNIC の選択的な移行を実行するには、移動する VNIC を指定します。次の例では、選択された VNIC を `net0` から `net1` に移動します。

```
# dladm modify-vnic -l net1 vnic0,vnic1,vnic2
```

---

注--L オプションは、グローバルな変更のみに制限されています。

---

VNIC のグループを移行しているときに、同時にそれらの VLAN ID も変更できます。ただし、新しい VLAN ID を割り当てるには、次の例に示すように VNIC を一度に 1 つずつ移行する必要があります。

```
# dladm modify-vnic -l net1 -v 123 vnic0
# dladm modify-vnic -l net1 -v 456 vnic1
# dladm modify-vnic -l net1 -v 789 vnic2
```

移行の MAC アドレスへの影響は、VNIC がソースデータリンクのファクトリ MAC アドレスを使用しているかどうかによって異なります。

- 移行中に `-m` オプションを指定しない場合は、移行のあと、ファクトリ MAC アドレスが、ターゲットデータリンクからランダムに割り当てられたアドレスに置き換えられます。
- 移行中に `-m address` オプションを使用した場合は、移行のあと、そのアドレスが VNIC に割り当てられます。

ランダムに割り当てられた MAC アドレスは影響を受けず、移行のあと、それらの各 VNIC によって保持されます。

次の例は、複数の VNIC を移行する方法を示しています。VNIC が、ランダムに割り当てられた MAC アドレスを使用していることに注意してください。そのため、移行のあと、これらのアドレスは変更されません。

```
# dladm show-vnic
LINK      OVER    SPEED  MACADDRESS      MACADDRTYPE  VID
vnic1     net0    1000   2:8:20:c2:39:38  random        0
vnic2     net0    1000   2:8:20:5f:84:ff  random        0
```

```
# dladm modify-vnic -l net1 -L net0
# dladm show-vnic vnic0
LINK      OVER      SPEED     MACADDRESS      MACADDRTYPE     VID
vnic1     net1     1000     2:8:20:c2:39:38  random          0
vnic2     net1     1000     2:8:20:5f:84:ff  random          0
```

## VNIC情報の表示

システム上のVNICに関する情報を取得するには、`dladm show-vnic` コマンドを使用します。

```
# dladm show-vnic
LINK      OVER      SPEED     MACADDRESS      MACADDRTYPE
vnic1     net0     1000 Mbps  2:8:20:c2:39:38  random
vnic2     net0     1000 Mbps  2:8:20:5f:84:ff  random
```

VNICはまた、データリンクでもあります。そのため、VNICがシステム上に存在する場合は、データリンクに関する情報を表示する任意の`dladm`コマンドを使用して、それらのVNICに関する情報を含めることもできます。たとえば、`dladm show-link`の一覧にはVNICが含まれます。または、`dladm show-linkprop`コマンドを使用してVNICのプロパティを確認することができます。1つのVNICに関するプロパティ情報を取得するには、リンクプロパティを表示するときにVNICを指定します。

```
# dladm show-linkprop [-p property] vnic
```

## ▼ VNICを削除する方法

この手順では、システムからVNIC構成を削除する方法について説明します。この手順では、VNICがゾーンに接続されていることを前提としています。この手順を実行するには、大域ゾーン内に存在している必要があります。

- 1 VNICがゾーンに接続されているため、そのゾーンを停止します。

```
global# zoneadm -z zone halt
```

---

注 - ゾーンによって使用されているリンクを特定するには、`dladm show-link` コマンドを使用します。

---

- 2 ゾーンからVNICを削除するか、または切り離します。

```
global# zonecfg -z zone remove net physical=vnic
```

- 3 システムからVNICを削除します。

```
global# dladm delete-vnic vnic
```

- 4 ゾーンをリブートします。

```
global# zonecfg -z zone boot
```

#### 例 2-5 システムからのVNICの削除

この例では、zoneB とシステムから vnic1 が削除されます。

```
Global# dladm show-link
LINK          CLASS  MTU   STATE  OVER
net0          phys   1500  up     --
net2          phys   1500  up     --
net1          phys   1500  up     --
net3          phys   1500  up     --
zoneA/net0    vnic   1500  up     net0
zoneB/net0    vnic   1500  up     net0
vnic0         vnic   1500  up     net1
zoneA/vnic0   vnic   1500  up     net1
vnic1         vnic   1500  up     net1
zoneB/vnic1   vnic   1500  up     net1

Global# zoneadm -z zoneB halt
Global# zonecfg -z zoneB remove net physical=vnic1
Global# dladm delete-vnic vnic1
Global# zonecfg -z zoneB reboot
```



# Oracle Solaris でのネットワークリソースの管理

---

この章では、データリンク (VNIC などの仮想リンクを含む) 上のリソースを管理する方法について説明します。ネットワークリソースの管理では、特に仮想ネットワークのパフォーマンスを向上させるために、IP サービスの品質 (QoS) を実装します。

この章の内容は次のとおりです。

- 35 ページの「クライアント、送信リング、および受信リングの操作」
- 45 ページの「プールおよび CPU の操作」
- 50 ページの「フロー上のリソースの管理」

## クライアント、送信リング、および受信リングの操作

NIC 上にある受信 (Rx) リングと送信 (Tx) リングはそれぞれ、システムがネットワークパケットの受信と送信を行うために使用するハードウェアリソースです。以降のセクションでは、リングの概要について説明したあと、ネットワークプロセスにリングを割り当てるために使用される手順を示します。また、リングを割り当てるコマンドを発行したときに、このメカニズムがどのように動作するかを示す例も提供されます。

## MAC クライアントとリング割り当て

VNIC やその他のデータリンクなどの MAC クライアントは、システムとほかのネットワークノードの間の通信を可能にするために NIC 上に構成されます。各クライアントは構成されたあと、Rx リングと Tx リングの両方を使用して、それぞれネットワークパケットを受信または送信します。MAC クライアントは、ハードウェアベースまたはソフトウェアベースのどちらかです。ハードウェアベースのクライアントは、次の条件のいずれかを満たします。

- 1 つ以上の Rx リングを専用に使用できます。

- 1つ以上の Tx リングを専用に使用できます。
- 1つ以上の Rx リングと1つ以上の Tx リングを専用に使用できます。

これらのいずれの条件も満たさないクライアントは、ソフトウェアベースの MAC クライアントです。

NICによっては、ハードウェアベースのクライアントにリングを排他的使用のために割り当てることができます。nxgeなどのNICは、動的なリング割り当てをサポートしています。このようなNICでは、ハードウェアベースのクライアントを構成できるわけではありません。リングが割り当てに使用できる状態のままにあると仮定すると、このようなクライアントに割り当てられるリングの数を決定できる柔軟性もあります。リングの使用は、常にプライマリインタフェース(たとえば、net0)のために最適化されます。プライマリインタフェースは、プライマリクライアントとも呼ばれます。ほかのクライアントによって排他的使用のために割り当てられていない使用可能なリングはすべて、プライマリインタフェースに自動的に割り当てられます。

ixgeなどのほかのNICは、静的なリング割り当てのみをサポートしています。これらのNICでは、ハードウェアベースのクライアントだけを作成できます。各クライアントには、クライアントごとに固定されたリングのセットが自動的に構成されます。この固定されたセットは、NICドライバの初期構成中に決定されます。静的なリング割り当てのためのドライバの初期構成についての詳細は、『[Oracle Solaris 11.1 カーネルのチューンアップ・リファレンスマニュアル](#)』を参照してください。

ソフトウェアベースのクライアントは、リングを排他的に使用できません。代わりに、これらのクライアントは、ほかの既存のソフトウェアベースのクライアントまたはプライマリクライアントとリングを共有します。ソフトウェアベースのクライアントが使用するリングは、リング割り当てで優先順位を持つハードウェアベースのクライアントの数によって異なります。

プライマリクライアントとその他のセカンダリクライアントの区別を理解することが重要です。プライマリクライアントは、NICの物理的なデータリンクです。インストール中にOracle Solarisによって提供される汎用名に基づいて、プライマリクライアントにはnetNという名前が付けられます。ここで、Nはインスタンス番号です。データリンクの汎用名については、『[Oracle Solaris 11 ネットワーキングの紹介](#)』の「[ネットワークデバイスとデータリンク名](#)」を参照してください。VNICは、物理的なデータリンク上に作成されるセカンダリクライアントです。これらのクライアントがハードウェアベースのクライアントである場合は、リングを排他的に使用できます。それ以外の場合、これらのクライアントはソフトウェアベースです。

## VLANでのリング割り当て

VLANでのリングの割り当て処理は、そのVLANが作成された方法によって異なります。VLANは、次の2つの方法のどちらかで作成されます。

- `dladm create-vlan` サブコマンドを使用して:

```
# dladm create-vlan -l link -v vid vlan
```

- `dladm create-vnic` サブコマンドを使用して:

```
# dladm create-vnic -l link -v vid vnic
```

`dladm create-vlan` サブコマンドで作成された VLAN は、ベースとなるインタフェースと同じ MAC アドレスを共有します。その結果、その VLAN は、ベースとなるインタフェースの Rx リングと Tx リングも共有します。`dladm create-vnic` コマンドを使用して VNIC として作成された VLAN には、ベースとなるインタフェースとは異なる MAC アドレスが割り当てられます。このような VLAN へのリングの割り当ては、ベースとなるリンクへの割り当てとは独立して行われます。そのため、NIC がハードウェアベースのクライアントをサポートしていると仮定すると、その VLAN には独自の専用リングを割り当てることができます。

## リング割り当てのためのデータリンクプロパティ

リングを管理するために、`dladm` コマンドを使用して、次の 2 つのリングプロパティを設定できます。

- `rxrings` は、指定されたリンクに割り当てられた Rx リングの数を示します。
- `txrings` は、指定されたリンクに割り当てられた Tx リングの数を示します。

各プロパティを次の 3 つの指定可能な値のいずれかに設定できます。

- `sw` は、ソフトウェアベースのクライアントを構成していることを示します。このクライアントは、リングを排他的に使用できません。代わりに、このクライアントは、同様に構成されたほかのすべての既存のクライアントとリングを共有します。
- $n > 0$  (0 より大きい数) は、ハードウェアベースのクライアントの構成にのみ適用されます。この数は、クライアントに排他的使用のために割り当てられるリングの数量を示します。数を指定できるのは、ベースとなる NIC が動的なリング割り当てをサポートしている場合だけです。
- `hw` もまた、ハードウェアベースのクライアントの構成に適用されます。ただし、このようなクライアントでは、専用リングの実際の数を指定することはできません。代わりに、NIC ドライバの初期構成に従って、クライアントごとに固定された数のリングがすでに設定されています。`*rings` プロパティを `hw` に設定するのは、ベースとなる NIC が静的なリング割り当てをサポートしている場合だけです。

現在のリングの割り当てと使用に関する情報を表示するために、次の追加の読み取り専用のリングプロパティが使用できます。

- `rxrings-available` と `txrings-available` は、割り当てに使用できる Rx リングと Tx リングの数を示します。

- `rxhwclnt-available`と`txhwclnt-available`は、NIC上に構成できるRxとTxのハードウェアベースのクライアントの数を示します。

## 受信および送信リングを操作するためのコマンド

データリンクの受信および送信リングの使用を管理するには、次の主体 `dladm` サブコマンドを使用します。

- `dladm show-linkprop` – リンクプロパティの現在の値を表示します (Rx および Tx リングを含む)。この出力では、データリンクのリングサポート機能に関する次の情報が提供されます。Rx および Tx リングを使用するように構成できるクライアントのタイプを判定するには、これらの情報が必要です。
  - 作成できる使用可能なクライアント
  - 使用可能なクライアントに割り当てることのできる使用可能なリング
  - 動的または静的なリング割り当てをサポートするための機能
  - 静的なリング割り当てのみがサポートされている場合は、既存のクライアントへのリングの現在の配布

このコマンドの出力を解釈する方法についての詳細は、[37 ページの「リング割り当てのためのデータリンクプロパティ」](#)を参照してください。

- `dladm show-phys -H datalink` – 物理的なデータリンクのリングが現在、既存のクライアントによってどのように使用されているかを表示します。
- `dladm create-vnic -p ring-properties vnic` – トラフィックの処理に使用するための特定の数の Tx または Rx リングを持つクライアントを作成します。
- `dladm set-linkprop -p ring-properties datalink` – リングが使用可能であり、かつリング割り当てがサポートされている場合は、特定のクライアントにリングを割り当てます。

## リング情報の取得および解釈

このセクションでは、データリンクのリング関連のプロパティを表示する `dladm show-linkprop` 出力について説明します。

### データリンクのリング割り当て機能の表示

このセクションでは、リング関連のプロパティに関するコマンド出力の例を示したあと、取得できる情報の種類について説明します。これらの例では、次の NIC が使用されています。

- `net0` (nxge 上)
- `net1` (ixgbe 上)

### ■ net2 (e1000g 上)

#### 例3-1 nxgeのリング情報

次の例は、nxgeのリング情報を示しています。

```
# dladm show-linkprop net0
LINK      PROPERTY          PERM  VALUE  DEFAULT  POSSIBLE
...
net0      rxrings           rw    --    --       sw,<1-7>
...
net0      txrings           rw    --    --       sw,<1-7>
...
net0      rxrings-available r-    5     --       --
net0      txrings-available r-    5     --       --
net0      rxhwclnt-available r-    2     --       --
net0      txhwclnt-available r-    2     --       --
...
```

net0では、POSSIBLEフィールドの値は、rxringsとtxringsに対してswと<1-7>です。これらの値は、nxgeがソフトウェアベースのクライアントだけでなく、ハードウェアベースのクライアントをサポートしていることを示します。範囲<1-7>は、クライアントに対して設定できるRxリングまたはTxリングの数の制限を示しています。この範囲はまた、NICが受信側と送信側の両方に対して動的なリング割り当てをサポートしていることも示しています。

さらに、\*rings-availableプロパティは、ハードウェアベースのクライアントに割り当てるために5つのRxリングと5つのTxリングが使用可能なことを示しています。

ただし、\*clnt-availableプロパティは、使用可能なRxリングを排他的に使用できるクライアントを2つしか構成できないことを示しています。同様に、使用可能なTxリングを排他的に使用できるクライアントは2つしか構成できません。

#### 例3-2 ixgbeのリング情報

次の例は、ixgbeのリング情報を示しています。

```
# dladm show-linkprop net1
LINK      PROPERTY          PERM  VALUE  DEFAULT  POSSIBLE
...
net1      rxrings           rw    --    --       sw,hw
...
net1      txrings           rw    --    --       sw,hw,<1-7>
...
net1      rxrings-available r-    0     --       --
net1      txrings-available r-    5     --       --
net1      rxhwclnt-available r-    0     --       --
net1      txhwclnt-available r-    7     --       --
...
```

net1では、rxringsとtxringsの両方に対するPOSSIBLEフィールドの値swとhwは、ixgbeがハードウェアベースのクライアントとソフトウェアベースのクライアント

## 例 3-2 ixgbe のリング情報 (続き)

トの両方をサポートしていることを示しています。Rx リングでは、静的なリング割り当てのみがサポートされています。つまり、ハードウェアがハードウェアベースの各クライアントに、固定された Rx リングのセットを割り当てます。ただし、Tx リングの場合、範囲 <1-7> は、動的割り振りがサポートされていることを示しています。ハードウェアベースのクライアントに割り当てられる Tx リングの数 (この例では、最大 7 リング) を判定できます。

さらに、\*rings-available プロパティは、ハードウェアベースのクライアントに割り当てられるために 5 つの Tx リングが使用可能であるが、割り当て可能な Rx リングはないことを示しています。

最後に、\*hwcnt-available プロパティに基づいて、ハードウェアベースの 7 つの Tx クライアントを Tx リングを排他的に使用するように構成できます。ただし、動的な Rx リング割り当てがサポートされていないため、Rx リングを排他的に使用できるハードウェアベースのクライアントを作成することはできません。

どちらかの \*rings-available プロパティの VALUE フィールドにあるゼロ (0) は、次のどちらかを示すことがあります。

- クライアントに割り当てられるために使用可能なリングがなくなった。
- 動的なリング割り当てがサポートされていない。

rxrings および txrings の POSSIBLE フィールドと、rxrings-available および txrings-available の VALUE フィールドと比較することによって、この 0 の意味を確認できます。

たとえば、次のように txrings-available が 0 であるとしします。

```
# dladm show-linkprop net1
LINK  PROPERTY          PERM  VALUE  DEFAULT  POSSIBLE
...
net1  rxrings           rw    --    --       sw,hw
net1  txrings           rw    --    --       sw,hw,<1-7>
net1  rxrings-available r-    0     --       --
net1  txrings-available r-    0     --       --
...
```

この出力では、rxrings-available の VALUE フィールドが 0 であるのに対して、rxrings の POSSIBLE フィールドは sw,hw です。これらの情報を組み合わせると、使用可能な Rx リングがないのは NIC が動的なリング割り当てをサポートしていないためであることがわかります。送信側では、txrings-available の VALUE フィールドが 0 であるのに対して、txrings の POSSIBLE フィールドは sw,hw,<1-7> です。これらの情報を組み合わせると、使用可能な Tx リングがないのはすべての Tx リングがすでに割り当てられているためであることがわかります。ただし、txrings の POSSIBLE フィールドが示しているように、動的なリング割り当てはサポートされています。そのため、Tx リングが使用可能になったら、これらのリングを割り当て

## 例 3-2 ixgbe のリング情報 (続き)

ることができます。

## 例 3-3 e1000g のリング情報

次の例は、e1000g のリング情報を示しています。

```
# dladm show-linkprop net2
LINK  PROPERTY                PERM  VALUE  DEFAULT  POSSIBLE
...
net2  rxrings                  rw    --    --    --
...
net2  txrings                  rw    --    --    --
...
net2  rxrings-available       r-    0     --    --
net2  txrings-available       r-    0     --    --
net2  rxhwcCnt-available      r-    0     --    --
net2  txhwcCnt-available      r-    0     --    --
...
```

この出力は、e1000g ではリング割り当てがサポートされていないため、リングもハードウェアベースのクライアントも構成できないことを示しています。

## データリンク上でのリングの使用とリングの割り当ての表示

次の2つの読み取り専用のデータリンクプロパティによって、リングが現在、データリンク上の既存のクライアントによってどのように使用されているかに関する情報が提供されます。

- rxrings-effective
- txrings-effective

リングの使用と、どのリングがクライアントに配布されているかに関する情報を取得するには、`dladm show-linkprop` サブコマンドと `dladm show-phys -H` サブコマンドの両方を使用します。

次の例は、Rx および Tx リングの使用と、これらのリングがクライアント間でどのように配布されているかに関して、この2つのコマンドによって生成された異なるタイプの出力を示しています。

## 例 3-4 プライマリクライアントのリングの使用

プライマリクライアントは、NIC の物理的なデータリンク上に構成されたインタフェースです。この例の場合、NIC は ixgbe カードです。デフォルトでは、そのデータリンクは `net0` です。net0 上の IP インタフェースはプライマリクライアントです。

```
# dladm show-linkprop net0
LINK  PROPERTY                PERM  VALUE  DEFAULT  POSSIBLE
...
```

例 3-4 プライマリクライアントのリングの使用 (続き)

```

net0  rxrings          rw  --  --  sw,hw
net0  rxrings-effective r-  2  --  --
net0  txrings          rw  --  --  sw,hw,<1-7>
net0  txrings-effective r   8  --  --
net0  txrings-available r-  7  --  --
net0  rxrings-available r-  0  --  --
net0  rxhwclnt-available r-  3  --  --
net0  txhwclnt-available r-  7  --  --
...
# dladm show-phys -H net0
LINK  RINGTYPE  RINGS  CLIENTS
net0  RX        0-1    <default,mcast>
net0  TX        0-7    <default>net0
net0  RX        2-3    net0
net0  RX        4-5    --
net0  RX        6-7    --

```

この出力では、プライマリクライアント net0 でのリングの使用と配布に関する次の情報が提供されます。

- rxrings-effective は、net0 が 2 つの Rx リングを自動的に受信することを示しています。txrings-effective は、net0 が 8 つの Tx リングを使用していることを示しています。デフォルトでは、未使用のリングはすべて、プライマリクライアントに自動的に割り当てられます。
- dladm show-phys -H コマンドの出力に基づくと、net0 に割り当てられた 2 つの Rx リングはリング 2 および 3 です。Tx リングに関しては、net0 はリング 0 から 7 を使用しています。

例 3-5 セカンダリクライアントのリングの使用

この例では、net0 上の VNIC クライアント vnic1 および ixgbe カードの物理的なデータリンクの構成を前提にしています。

```

# dladm show-linkprop vnic1
LINK  PROPERTY  PERM  VALUE  DEFAULT  POSSIBLE
...
vnic1 rxrings          rw  hw  --  sw,hw
vnic1 rxrings-effective r-  2  --  --
vnic1 txrings          rw  hw  --  sw,hw,<1-7>
vnic1 txrings-effective r-  1  --  --
...
# dladm show-linkprop net0
LINK  PROPERTY  PERM  VALUE  DEFAULT  POSSIBLE
...
net0  rxrings          rw  --  --  sw,hw
net0  rxrings-effective r-  2  --  --
net0  txrings          rw  --  --  sw,hw,<1-7>
net0  txrings-effective r-  --  --  --
net0  txrings-available r-  6  --  --
net0  rxrings-available r-  0  --  --
net0  rxhwclnt-available r-  3  --  --
net0  txhwclnt-available r-  6  --  --

```

## 例 3-5 セカンダリクライアントのリングの使用 (続き)

```

...
# dladm show-phys -H net0
LINK      RINGTYPE  RINGS      CLIENTS
net0     RX        0-1        <default,mcast>
net0     TX        0,2-7      <default>net0
net0     RX        2-3        net0
net0     RX        4-5        vnic1
net0     RX        6-7        --
net0     TX        1          vnic1

```

これらの3つのコマンドの出力を組み合わせると、次の情報が提供されます。

- vnic1 の rxrings-effective は、この VNIC が2つの Rx リングを自動的に受信することを示しています。txrings-effective は、vnic1 が1つの Tx リングを使用していることを示しています。\*ring プロパティに対して設定された hw 値で示されているように、これらのリングは静的に割り当てられます。
- dladm show-phys -H コマンドの出力に基づくと、net0 に割り当てられた2つの Rx リングはリング2および3です。Tx リングに関しては、net0 はリング0とリング2から7を使用しています。vnic1 は、Tx リングにはリング1を、Rx リングにはリング4および5を使用しています。

vnic1 は、静的なリング割り当てでハードウェアベースのクライアントとして構成されていることに注意してください。その結果、net0 上に作成できる使用可能な Tx ハードウェアクライアント (txhwcInt-available) の数は6に削減されています。

## ▼ クライアントを構成してリングを割り当てる方法

この手順では、リング割り当てに対するサポートの種類に基づいて、データリンク上にクライアントを構成する方法について説明します。38 ページの「データリンクのリング割り当て機能の表示」および41 ページの「データリンク上でのリングの使用とリングの割り当ての表示」で説明されているように、データリンクのリングプロパティを表示する dladm コマンドの出力を解釈できることを確認してください。これらの情報は、クライアントを構成するうえで役立ちます。

### 1 データリンクのリングプロパティを表示します。

```
# dladm show-linkprop datalink
```

この出力から、次の内容を判定します。

- NIC がハードウェアベースのクライアントをサポートしているかどうか
- NIC がサポートしているリング割り当てのタイプ
- ハードウェアベースのクライアントに割り当てするためのリングの可用性
- リンク上に構成できるハードウェアベースのクライアントの可用性

### 2 前の手順で得られた情報に応じて、次のいずれかを実行します。

- NICが動的なリング割り当てをサポートしている場合は、次の構文を使用してハードウェアベースのクライアントを作成します。

```
# dladm create-vnic -p rxrings=number[,txrings=number] -l link vnic
```

このクライアントが以前に作成されている場合は、次の構文を使用します。

```
# dladm set-linkprop -p rxrings=number[,txrings=number] vnic
```

---

注-NICによっては、RxリングまたはTxリングのどちらかで動的なリング割り当てをサポートしているが、両方のタイプではサポートしていないものがあります。*number*は、動的なリング割り当てがサポートされているリングのタイプに対して指定してください。

---

- NICが静的なリング割り当てをサポートしている場合は、次の構文を使用してハードウェアベースのクライアントを作成します。

```
# dladm create-vnic -p rxrings=hw[,txrings=hw] -l link vnic
```

このクライアントが以前に作成されている場合は、次の構文を使用します。

```
# dladm set-linkprop -p rxrings=hw[,txrings=hw] vnic
```

---

注-NICによっては、RxリングまたはTxリングのどちらかで静的なリング割り当てをサポートしているが、両方のタイプではサポートしていないものがあります。*hw*は、静的なリング割り当てがサポートされているリングのタイプに対して指定してください。

---

- NICがソフトウェアベースのクライアントのみをサポートしている場合は、次の構文を使用してクライアントを作成します。

```
# dladm create-vnic -p rxrings=sw[,txrings=sw] -l link vnic
```

このクライアントが以前に作成されている場合は、次の構文を使用します。

```
# dladm set-linkprop -p rxrings=sw[,txrings=sw] vnic
```

- 3 (オプション)新しく作成されたクライアントのリング情報を確認します。

```
# dladm show-linkprop vnic
```

- 4 (オプション)データリンクのリングが異なるクライアントの間でどのように配布されているかを確認します。

```
# dladm show-phys -H datalink
```

参照 仮想ネットワーク内のネットワークトラフィックを処理するためにフローを使用する方法や、システムリソース (RxおよびTxリングを含む) を割り当てる方法を示す例については、[例 3-8](#)を参照してください。

## プールおよびCPUの操作

`pool` リンクプロパティを使用すると、ネットワーク処理をCPUのプールにバインドできます。このプロパティを使用すると、ネットワークリソースの管理を、ゾーン内のCPUバインディングや管理により適切に統合できます。Oracle Solaris でのゾーン管理には、`zonecfg` または `poolcfg` コマンドを使用して、ネットワーク以外のプロセスをCPUリソースのプールにバインドする処理が含まれます。その同じリソースのプールをネットワークプロセスも管理するように専用に割り当てるには、`dladm set-linkprop` コマンドを使用してリンクの `pool` プロパティを構成します。次に、ゾーンにそのリンクを割り当てます。

リンクの `pool` プロパティを設定し、そのリンクをゾーンのネットワークインタフェースとして割り当てることによって、そのリンクがゾーンのプールにもバインドされます。そのゾーンが排他的ゾーンになるように設定されると、プール内のCPUリソースは、そのゾーンに割り当てられていないほかのデータリンクから使用できなくなります。

---

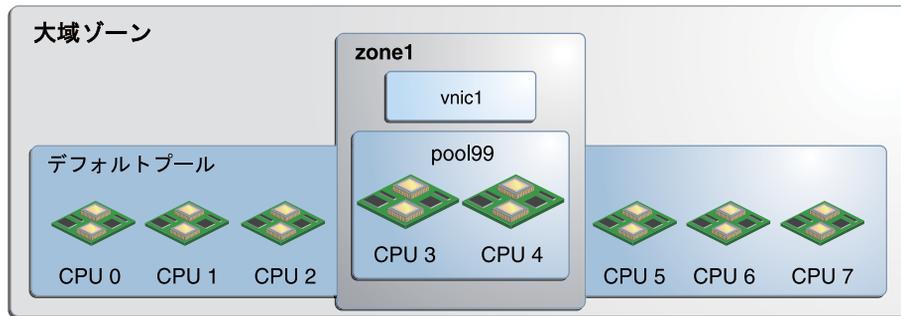
注-別のプロパティ `cpu` を設定すると、データリンクに特定のCPUを割り当てることができます。この2つのプロパティ `cpu` と `pool` は相互に排他的です。特定のデータリンク用に両方のプロパティを設定することはできません。`cpu` プロパティを使用してデータリンクにCPUリソースを割り当てるには、[49 ページの「リンクにCPUを割り当てる方法」](#)を参照してください。

---

ゾーン内のプールについての詳細は、『[Oracle Solaris 11.1 の管理: Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理](#)』の第13章「リソースプールの作成と管理(タスク)」を参照してください。プールの作成およびCPUセットのプールへの割り当てについての詳細は、`poolcfg(1M)` のマニュアルページを参照してください。

次の図は、データリンクに `pool` プロパティが割り当てられたときのプールの動作を示しています。

図 3-1 ゾーンに割り当てられた VNIC の pool プロパティ

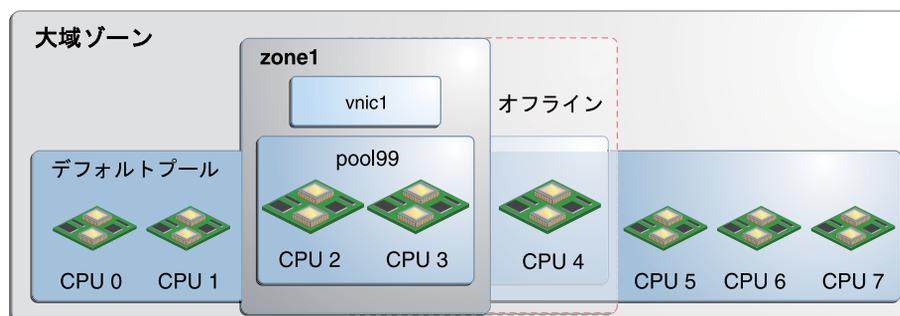


この図では、システムに8つのCPUが割り当てられています。システム上にプールが構成されていない場合は、すべてのCPUがデフォルトプールに属し、大域ゾーンで使用されます。ただし、この例では、CPU 3とCPU 4で構成された pool99 プールが作成されています。このプールは、排他的ゾーンである zone1に関連付けられています。pool99が vnic1のプロパティとして設定された場合、pool99は vnic1のネットワークプロセスの管理にも専用で使用されます。vnic1が zone1のネットワークインタフェースとして割り当てられたあと、pool99内のCPUは、zone1のネットワークプロセスとネットワーク以外のプロセスの両方を管理するために予約されます。

pool プロパティは本質的に動的で、ゾーンのプールには一連のCPUを構成することができ、そのプールのCPUセットにどのCPUが割り当てられるかはカーネルによって決定されます。プールに対する変更はデータリンク用に自動的に実装されるため、そのリンク用のプールの管理が単純化されます。これに対して、cpu プロパティを使用してリンクに特定のCPUを割り当てるには、割り当てられるCPUを指定する必要があります。プールのCPUコンポーネントを変更しようとするたびにcpu プロパティを設定する必要があります。

たとえば、図 3-1 のシステム CPU 4 がオフラインになったとします。pool プロパティは動的であるため、ソフトウェアによって追加のCPUがプールに自動的に関連付けられます。そのため、2CPUというプールの元の構成が保持されます。vnic1にとって、この変更は透過的です。調整された構成を次の図に示します。

図 3-2 pool プロパティの自動再構成



プール関連の追加のプロパティによって、CPU または CPU のプールのデータリンクの使用に関する情報が表示されます。これらのプロパティは読み取り専用であり、管理者が設定することはできません。

- `pool-effective` は、ネットワークプロセスに使用されているプールを表示します。
- `cpus-effective` は、ネットワークプロセスに使用されている CPU の一覧を表示します。

ゾーンの CPU リソースを管理するために、初期手順として、データリンクの `pool` プロパティが設定されることは通常ありません。より一般的な方法として、リソースのプールを使用するようにゾーンを構成するために `zonecfg` や `poolcfg` などのコマンドが使用されます。`cpu` や `pool` のリンクプロパティ自体が設定されることはありません。その場合、これらのデータリンクの `pool-effective` および `cpus-effective` プロパティは、ゾーンがブートされたときのゾーン構成に従って自動的に設定されます。`pool-effective` ではデフォルトプールが表示され、`cpus-effective` の値はシステムによって選択されます。そのため、`dladm show-linkprop` コマンドを使用した場合、`pool` および `cpu` プロパティは空になりますが、`pool-effective` および `cpus-effective` プロパティには値が含まれます。

データリンクの `pool` および `cpu` プロパティの直接の設定は、ゾーンの CPU プールをネットワークプロセス用にバインドするために使用できる代替手順です。これらのプロパティを構成すると、それらの値が `pool-effective` および `cpus-effective` プロパティにも反映されます。ただし、この代替手順を使用してゾーンのネットワークリソースを管理することは通常あまり行われません。

## ▼ CPU プールをデータリンク用に構成する方法

ほかのリンクプロパティと同様に、`pool` プロパティも、そのリンクが作成された時点か、またはあとでそのリンクをさらに構成する必要が発生したときにデータリンクに対して設定できます。

VNICを作成するときに pool プロパティを設定するには、次の構文を使用します。

```
# dladm create-vnic -p pool=pool-name -l link vnic
```

既存の VNIC の pool プロパティを設定するには、次の構文を使用します。

```
# dladm setlinkprop -p pool=pool-name vnic
```

次の手順は、CPU プールを VNIC 用に構成する方法を説明しています。

始める前に 次の作業を完了している必要があります。

- 割り当てられた数の CPU を含むプロセッサセットを作成します。
- そのプロセッサセットを関連付けるプールを作成します。
- そのプールをプロセッサセットに関連付けます。

---

注-これらの前提条件を完了するための手順については、『Oracle Solaris 11.1の管理: Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理』の「構成の変更方法」を参照してください。

---

- 1 リンクの pool プロパティを、ゾーン用に作成した CPU のプールに設定します。VNIC が存在するかどうかに応じて、次の手順のいずれかを実行します。

- VNIC がまだ作成されていない場合は、次の構文を使用します。

```
# dladm create-vnic -l link -p pool=pool vnic
```

ここで、pool は、ゾーン用に作成されたプールの名前を示します。

- VNIC が存在する場合は、次の構文を使用します。

```
# dladm setlinkprop -p pool=pool vnic
```

- 2 ゾーンをその VNIC を使用するように設定します。

```
zonecfg>zoneid:net> set physical=vnic
```

---

注-ゾーンにネットワークインタフェースを割り当てる方法を説明した手順については、『Oracle Solaris 11.1の管理: Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理』の「ゾーンを構成、検証、および確定する」を参照してください。

---

### 例 3-6 リンクの CPU プールを排他的な IP タイプを持つゾーンに割り当てる

この例は、プールをゾーンのデータリンクに割り当てる方法を示しています。このシナリオは、図 3-1 の構成に基づいています。この例では、pool99 という名前の CPU のプールが、すでにゾーン用に構成されていることを前提にしています。この

プールが次に、VNICに割り当てられます。最後に、非大域ゾーン `zone1` が、このVNICをネットワークインタフェースとして使用するよう設定されます。

```
# dladm create-vnic -l net1 -p pool99 vnic1

# zonecfg -c zone1
zonecfg:zone1> set ip-type=exclusive
zonecfg:zone1> add net
zonecfg:zone1>net> set physical=vnic1
zonecfg:zone1>net> end
zonecfg:zone1> exit
```

## ▼ リンクにCPUを割り当てる方法

次の手順は、cpuプロパティを構成することによって、データリンクをたどっているトラフィックの処理に特定のCPUを割り当てる方法を説明しています。

- 1 インタフェースのCPU割り当てを確認します。

```
# dladm show-linkprop -p cpus link
```

デフォルトでは、特定のどのインタフェースにもCPUは割り当てられていません。そのため、コマンド出力にあるパラメータVALUEにはどのエントリも含まれません。

- 2 各割り込みと、それらの割り込みが関連付けられているCPUを一覧表示します。

```
# echo ::interrupts | mdb -k
```

この出力には、システム上のリンクごとのパラメータ (CPU番号を含む) が一覧表示されます。

- 3 リンクにCPUを割り当てます。

このCPUには、そのリンクの割り込みが関連付けられているCPUを含めることができます。

```
# dladm set-linkprop -p cpus=cpu1,cpu2,... link
```

ここで、`cpu1` は、そのリンクに割り当てるCPU番号です。そのリンク専用に複数のCPUを割り当てることができます。

- 4 リンクの割り込みをチェックして、新しいCPU割り当てを確認します。

```
# echo ::interrupts | mdb -k
```

- 5 (オプション) リンクに関連付けられたCPUを表示します。

```
# dladm show-linkprop -p cpus link
```

### 例 3-7 リンクへのCPUの割り当て

この例は、データリンク net0 に特定の CPU を専用に割り当てる方法を示しています。

さまざまなコマンドで生成される出力にある次の情報に注目してください。わかりやすくするために、出力にある重要な情報が強調されています。

- デフォルトでは、net0 に専用の CPU は割り当てられていません。そのため、VALUE は -- です。
- net0 の割り込みは、CPU 18 に関連付けられています。
- CPU が割り当てられたあと、net0 の VALUE には新しい CPU の一覧が表示されません。

```
# dladm show-linkprop -p cpus net0
LINK      PROPERTY  PERM  VALUE      DEFAULT  POSSIBLE
net0      cpus      rw    --         --       --

# echo ::interrupts | mdb -k
Device  Shared  Type  MSG #  State  INO   Mondo  Pil  CPU
net#0   no      MSI   2      enbl   0x1a  0x1a   6    18

# dladm set-linkprop -p cpus=14,18,19,20 net0

# dladm show-linkprop -p cpus net0
LINK      PROPERTY  PERM  VALUE      DEFAULT  POSSIBLE
net0      cpus      rw    14,18,19,20  --       --
```

割り込みを含め、サポートしているスレッドはすべて、新しく割り当てられた一連の CPU に限定されるようになりました。

**参照** 仮想ネットワーク内のネットワークトラフィックを処理するためにフローを使用する方法や、システムリソース (CPU および CPU プールを含む) を割り当てる方法を示す例については、[例 3-8](#) を参照してください。

## フロー上のリソースの管理

フローは、属性に従って構成されたネットワークパケットから成ります。フローを使用すると、ネットワークリソースをさらに詳細に割り当てることができます。フローの概要については、[13 ページの「フローを使用することによるネットワークリソースの管理」](#)を参照してください。

リソースの管理にフローを使用するには、次の一般的な手順を実行します。

1. [13 ページの「フローを使用することによるネットワークリソースの管理」](#) に示されている特定の属性に基づいてフローを作成します。

2. ネットワークリソースに関連するプロパティを設定することによって、リソースのフローの使用をカスタマイズします。現在、設定できるのはパケットを処理するための帯域幅だけです。

## ▼ フローを構成する方法

- 1 必要に応じて、フローを構成するリンクを決定するために、使用可能なリンクを一覧表示します。

```
# dladm show-link
```

- 2 選択されたリンク上の IP インタフェースに IP アドレスが正しく構成されていることを確認します。

```
# ipadm show-addr
```

- 3 フローごとに決定した属性に従ってフローを作成します。

```
# flowadm add-flow -l link -a attribute=value[,attribute=value] flow
```

*link* フローを構成しているリンクを示します。

*attribute* ネットワークパケットをフローに構成するときを使用できる次の分類のいずれかを示します。

- IP アドレス
- トランスポートプロトコル (UDP、TCP、または SCTP)
- アプリケーションのポート番号 (たとえば、FTP ではポート 21)
- DS フィールドの属性 (IPv6 パケットでのサービスの品質にのみ使用されます) DS フィールドについての詳細は、『[Oracle Solaris 11.1 での IP サービス品質の管理](#)』の「DS コードポイント」を参照してください。

*flow* 特定のフローに割り当てる名前を示します。

フローとフローの属性についての詳細は、[flowadm\(1M\)](#) のマニュアルページを参照してください。

- 4 (オプション) データリンクの帯域幅に対する値の可能性のある範囲を表示します。

```
# dladm show-linkprop -p maxbw link
```

ここで、*link* は、そのフローが構成されているデータリンクです。

値の範囲が POSSIBLE フィールドに一覧表示されます。

- 5 フローに帯域幅共有を割り当てます。

```
# flowadm set-flowprop -p maxbw=value flow
```

設定する値は、リンクの帯域幅に対して許可される値の範囲内になければなりません。

---

注- 現在、カスタマイズできるのはフローの帯域幅だけです。

---

- 6 (オプション) リンク上に作成したフローを表示します。

# **flowadm**

---

注- `flowadm` コマンド (サブコマンドなしで使用された場合) によって、`flowadm show-flow` コマンドと同じ情報が提供されます。

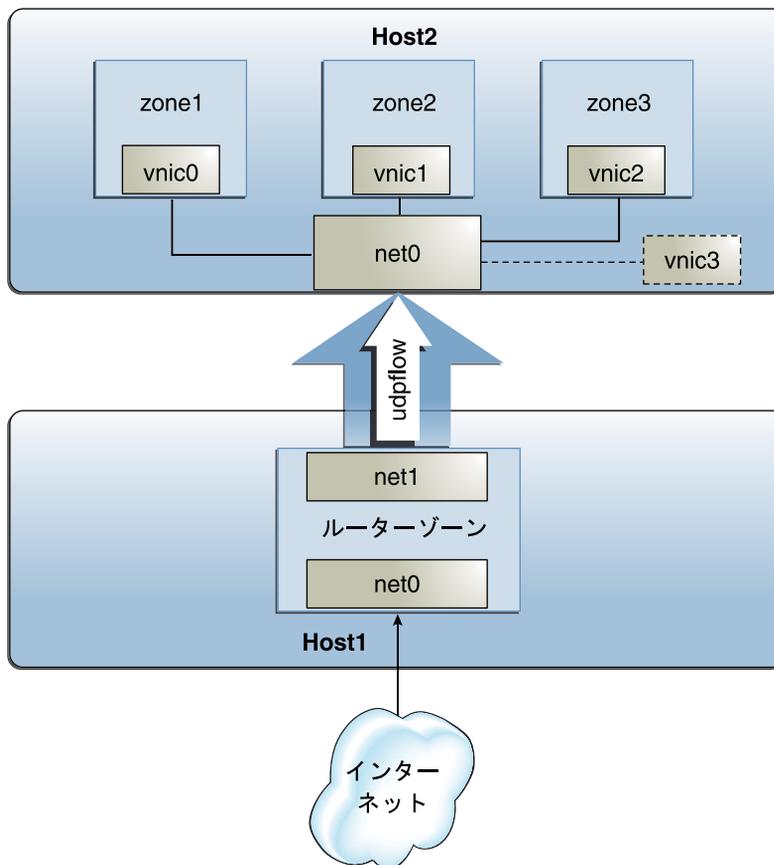
---

- 7 (オプション) 指定したフローのプロパティ値を表示します。

# **flowadm show-flowprop** *flow*

**例 3-8** リンクとフローのプロパティを設定することによるリソースの管理

この例では、データリンクとフローの両方にネットワークリソースを割り当てるための手順が結合されています。この例は、次の図に示す構成に基づいています。



この図は、互いに接続された2つの物理ホストを示しています。

- Host1の構成は次のとおりです。
  - ルーターゾーンとして機能する1つの非大域ゾーンが存在します。このゾーンには、次の2つのインターフェースが割り当てられています。net0はインターネットに接続し、net1は2番目のホストを含む内部ネットワークに接続します。
  - `udpfloor`は、UDPトラフィックを分離し、UDPパケットのリソースの使用方法に対する制御を実装するためにnet0上に構成されたフローです。フローの構成については、50ページの「フロー上のリソースの管理」を参照してください。
- Host2の構成は次のとおりです。

- 3つの非大域ゾーンと、それぞれ対応する VNIC が存在します。これらの VNIC は、カードが動的なリング割り当てをサポートする net0 上に構成されています。リング割り当てについての詳細は、35 ページの「クライアント、送信リング、および受信リングの操作」を参照してください。
- 各ゾーンのネットワーク処理の負荷は異なります。この例では、zone1 の負荷は高く、zone2 の負荷は中程度であり、zone3 の負荷は低いものとします。リソースは、それぞれの負荷に従ってこれらのゾーンに割り当てられています。
- 別の VNIC がソフトウェアベースのクライアントとして構成されています。MAC クライアントの概要については、35 ページの「MAC クライアントとリング割り当て」を参照してください。

この例でのタスクには、次のものが含まれます。

- フローの作成およびフロー制御の構成 - Host2 で受信される UDP パケットに対する個別のリソース制御を作成するために、net1 上にフローが作成されます。
- Host2 上の VNIC のためのネットワークリソースのプロパティの構成 - ゾーンごとの処理の負荷に基づいて、各ゾーンの VNIC に一連の専用リングが構成されます。また、ソフトウェアベースのクライアントの例として、専用リングのない別の VNIC も構成されます。

この例にはゾーン構成のための手順が含まれていないことに注意してください。ゾーンを構成するには、『Oracle Solaris 11.1 の管理: Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理』の第 17 章「非大域ゾーンの計画と構成(タスク)」を参照してください。

最初に、Host1 上のリンクと IP インタフェースに関する情報を表示します。

```
# ipadm
NAME                CLASS/TYPE STATE   UNDER  ADDR
lo0                  loopback  ok      --      --
  lo0/v4              static    ok      --      127.0.0.1/8
net0                  ip        ok      --      --
  net0/v4              static    ok      --      10.10.6.5/24
net1                  ip        failed  ipmp0   --
  net1/v4              static    ok      --      10.10.12.42/24
```

次に、Host2 への UDP トラフィックを分離するために、net1 上にフローを作成します。次に、そのフロー上にリソース制御を実装します。

```
# flowadm add-flow -l net1 -a transport=udp udpflow
# flowadm set-flowprop -p maxbw=80 udpflow
```

次に、作成されたフローに関する情報を確認します。

```
flowadm
FLOW      LINK  IPADDR  PROTO  LPORT  RPORT  DFSLD
udpflow   net1  --      udp    --     --     --
```

```
# flowadm show-flowprop
FLOW          PROPERTY          VALUE          DEFAULT          POSSIBLE
udpflow       maxbw              80             --              --
```

Host2 上で、ゾーンごとに net0 上に VNIC を構成します。各 VNIC 上にリソース制御を実装します。次に、各 VNIC をそれぞれ対応するゾーンに割り当てます。

```
# dladm create-vnic -l net0 vnic0
# dladm create-vnic -l net0 vnic1
# dladm create-vnic -l net0 vnic2
```

```
# dladm set-prop -p rxrings=4,txrings=4 vnic0
# dladm set-prop -p rxrings=2,txrings=2 vnic1
# dladm set-prop -p rxrings=1,txrings=1 vnic2
```

```
# zonecfg -z zone1
# zonecfg:zone1> add net
# zonecfg:zone1:net> set physical=vnic0
# zonecfg:zone1:net> end
# zonecfg:zone1> commit
# zonecfg:zone1> exit
# zoneadm -z zone1 reboot
```

```
# zonecfg -z zone2
# zonecfg:zone2> add net
# zonecfg:zone2:net> set physical=vnic1
# zonecfg:zone2:net> end
# zonecfg:zone2> commit
# zonecfg:zone2> exit
# zoneadm -z zone2 reboot
#
```

```
# zonecfg -z zone3
# zonecfg:zone3> add net
# zonecfg:zone3:net> set physical=vnic2
# zonecfg:zone3:net> end
# zonecfg:zone3> commit
# zonecfg:zone3> exit
# zoneadm -z zone3 reboot
#
```

Host2 内の一連の CPU である pool1 が、以前に zone1 で使用するよう構成されていたとします。次のように、その CPU のプールを zone1 のネットワークプロセスも管理するようにバインドします。

```
# dladm set-prop -p pool=pool1 vnic0
```

最後に、プライマリインタフェースである net0 とリングを共有するソフトウェアベースのクライアントを作成します。

```
# dladm create-vnic -p rxrings=sw,txrings=sw -l net0 vnic3
```



# Oracle Solaris でのネットワークトラフィックとリソース使用状況の監視

---

この章では、物理および仮想ネットワーク環境 Oracle Solaris 11 でのネットワークリソースの使用に関する統計情報を監視したり、収集したりするためのタスクについて説明します。これらの情報は、プロビジョニング、統合、および請求の目的でリソース割り当てを解析するのに役立ちます。この章では、`dlstat` と `flowstat` という、統計情報を表示するために使用する 2 つのコマンドについて説明します。

この章では、次の内容について説明します。

- 57 ページの「ネットワークトラフィックフローの概要」
- 60 ページの「トラフィックの統計情報を監視するためのコマンド」
- 60 ページの「リンク上のネットワークトラフィックに関する統計情報の収集」
- 65 ページの「フロー上のネットワークトラフィックに関する統計情報の収集」
- 67 ページの「ネットワークトラフィックのためのネットワークアカウントिंगの構成」

## ネットワークトラフィックフローの概要

パケットは、システムに入力されるか、またはシステムから出力されるときにパスをたどります。詳細なレベルで見ると、パケットは、NIC の受信 (Rx) リングおよび送信 (Tx) リングを通して送受信されます。これらのリングから、受信パケットはさらに処理するためにネットワークスタックに渡され、送信パケットはネットワークに送信されます。

このセクションでは、ネットワークレーンの概念を導入します。ネットワークレーンとは、ネットワークトラフィックを管理するために割り当てられたシステムリソースの組み合わせです。そのため、ネットワークレーンは、特定のタイプのネットワークトラフィックのためにカスタマイズされたパスです。各レーンは、ハードウェアレーンまたはソフトウェアレーンのどちらかにすることができます。さらに、各レーンのタイプを受信レーンまたは送信レーンのどちらかにすることができます。

ハードウェアレーンとソフトウェアレーンの区別は、35 ページの「クライアント、送信リング、および受信リングの操作」で説明されているように、リングとリング割り当てをサポートする NIC の機能に基づいています。この章では主に、受信レーンを通して受信される受信トラフィックについて重点的に説明します。

---

注- データリンクのプロパティを設定することによって、Rx および Tx リングやその他のネットワークリソースが割り当てられます。そのため、データリンクはシステム上のネットワークレーンです。

---

ハードウェアレーン上では、パケットは、そのレーンに割り当てられたリングを排他的に使用できます。これに対して、ソフトウェアレーン上のリングは、そのレーン上のすべてのネットワークパケットで共有されます。データリンクは、次の理由から、リングを共有するように構成されます。

- 管理の目的。データリンクが、専用リングを必要とするリソース集約型プロセスを実行していない可能性があります。
- NIC でリング割り当てがサポートされていない。
- リング割り当てがサポートされているにもかかわらず、リングを排他的使用のために割り当てることができなくなった。

次の図は、さまざまなハードウェアレーンを示しています。

図4-1 ハードウェアレーン

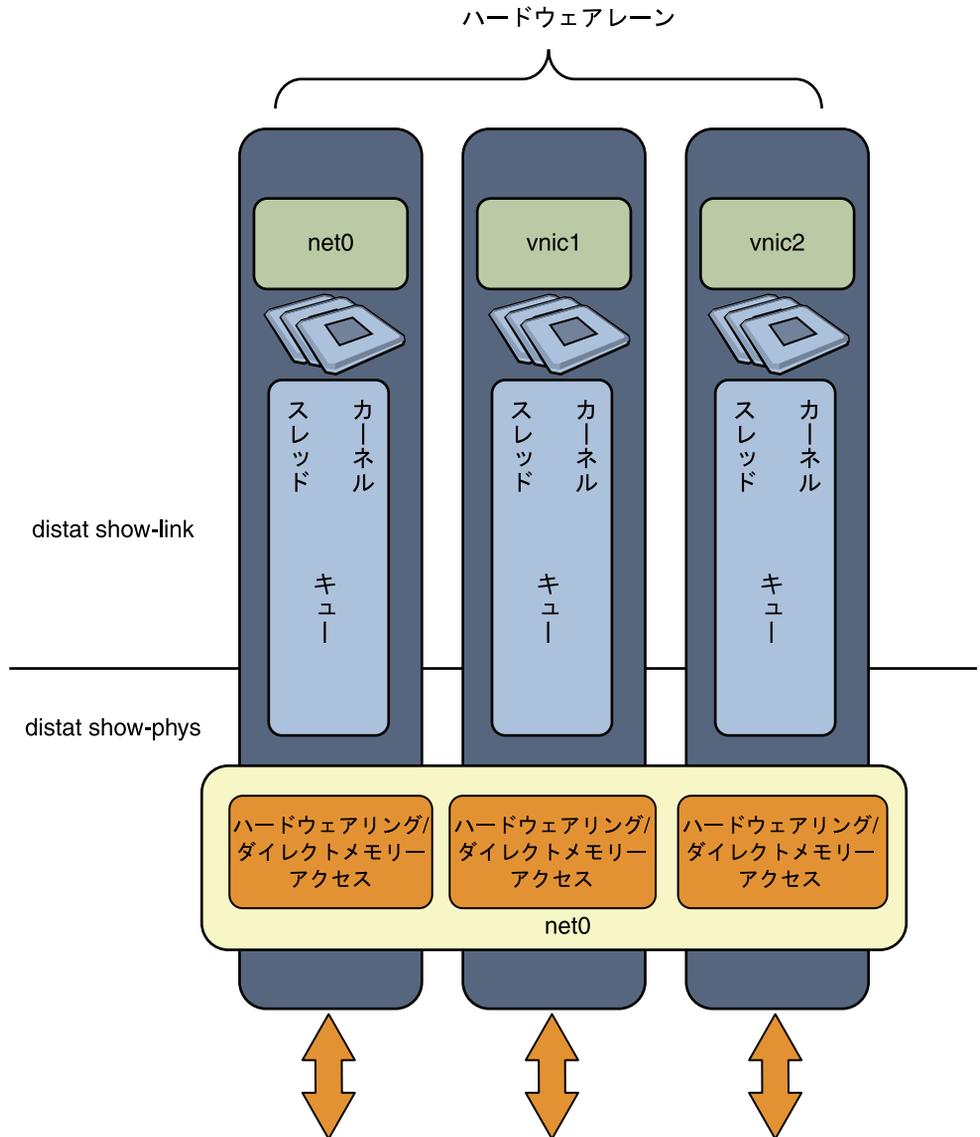


図4-1は、次の構成を示しています。

- net0 上には vnic1 と vnic2 の2つの VNIC が構成されています。プライマリクライアントとして、net0 はプライマリレーンでもあります。セカンダリクライアントとしてのこれらの VNIC もネットワークレーンです。

- セカンダリクライアントには、ハードウェアリングのセットが割り当てられています。各クライアント(プライマリクライアントを含む)は、ハードウェアレーンとして機能します。
- 各レーンには一連のCPUが割り当てられています。

以降のセクションでは、これらのレーンを通るトラフィックを監視する方法について説明します。

## トラフィックの統計情報を監視するためのコマンド

dlstat コマンドと flowstat コマンドでは、それぞれ、データリンクとフロー上のネットワークトラフィックに関する統計情報を監視および取得することができます。これらのコマンドは、dladm および flowadm コマンドと同等です。次の表に、\*adm コマンドのペアと \*stat コマンドのペアの機能比較を示します。

管理コマンド		監視コマンド	
コマンド	機能	コマンド	機能
dladm コマンドオプション	データリンクを構成および管理するためのユーザーインタフェースとツール	dlstat コマンドオプション	データリンク上のトラフィックに関する統計情報を取得するためのユーザーインタフェースとツール
flowadm コマンドオプション	フローを構成および管理するためのユーザーインタフェースとツール	flowstat コマンドオプション	フロー上のトラフィックに関する統計情報を取得するためのユーザーインタフェースとツール

以降のセクションでは、各監視コマンドについてさらに詳細に説明します。

## リンク上のネットワークトラフィックに関する統計情報の収集

dlstat コマンドの次のバリエーションを使用すると、ネットワークトラフィック情報を収集できます。

コマンド	提供される情報
<code>dlstat [link]</code> <code>dlstat -rt [link]</code> <code>dlstat show-link [link]</code>	レーンごとのインバウンドおよびアウトバウンドトラフィックの統計情報
<code>dlstat show-link -rt [link]</code>	リングごと、レーンごとのインバウンドおよびアウトバウンドトラフィックの統計情報
<code>dlstat show-phys [link]</code>	ネットワーク物理デバイスごとのインバウンドおよびアウトバウンドトラフィックの統計情報
<code>dlstat show-phys -rt [link]</code>	リングごと、ネットワーク物理デバイスごとのインバウンドおよびアウトバウンドトラフィックの統計情報
<code>dlstat show-aggr [link]</code> <code>dlstat show-aggr -rt [link]</code>	ポートごと、アグリゲーションごとのインバウンドおよびアウトバウンドトラフィックの統計情報

`dlstat` コマンドの `-r` オプションまたは `-t` オプションを使用すると、統計情報をそれぞれ、受信側または送信側に制限できます。さらに、`dlstat` コマンドのその他のオプションも使用できます。詳細は、[dlstat\(1M\)](#) のマニュアルページを参照してください。

## ネットワークデバイス上のネットワークトラフィックの統計情報の取得

`dlstat show-phys` サブコマンドは、物理ネットワークデバイスに関する統計情報を提供します。[図 4-1](#) に示すように、このサブコマンドは、ネットワークスタックのデバイス層にあるハードウェアリング上で動作します。また、同等のサブコマンド `dladm show-phys` も、このスタックの同じレベルで動作します。[図 4-1](#) を、『[Oracle Solaris 11 ネットワーキングの紹介](#)』の「[Oracle Solaris 11 の実装](#)」に示されているネットワークスタックと比較してください。

次の例は、システム上のすべての物理リンクに関する統計情報を示しています。

```
# dlstat show-phys
LINK  IPKTS  RBYTES  OPKTS   OBYTES
net0   2.14M  257.48M  3.19M   210.88M
net1   1.15M  120.32M  1.00M   98.70M
net2   1.10M  110.10M  1.28    183.00M
...
```

この出力は、システム上の各リンクに関する受信トラフィックと送信トラフィックの両方の統計情報を示しています。パケットの数とそれらのバイトサイズが表示されています。

次の例は、net0 の各ハードウェアリングに関する受信側の統計情報を示しています。

```
# dlstat show-phys -r net0
LINK  TYPE  ID  INDEX  IPKTS  RBYTES
net0   rx    local  --      0      0
net0   rx    hw     1      0      0
net0   rx    hw     2    1.73M  2.61G
net0   rx    hw     3      0      0
net0   rx    hw     4    8.44M  12.71G
net0   rx    hw     5    5.68M  8.56G
net0   rx    hw     6    4.99M  7.38G
net0   rx    hw     7      0      0
```

この2番目の出力では、net0 デバイスに8つの受信リングが割り当てられており、これは、INDEX フィールドで識別されます。リングごとのパケットの均一な分布は、それらのリングが、各リングの負荷に応じてリンクに正しく割り当てられていることを示す理想的な構成です。不均一な分布は、リンクごとのリングの不均衡な分布を示していることがあります。不均一な配布の解決は、NICが動的なリング割り当てをサポートしているかどうかによって異なります。サポートしている場合は、パケットをより均等に処理するように、リンクごとにリングを再配布できます。動的なリング割り当てについての詳細は、35 ページの「クライアント、送信リング、および受信リングの操作」を参照してください。

次の例は、デバイス上で毎秒受信されているトラフィックに関する情報を示しています。この間隔は、-i オプションを使用して指定されます。表示のリフレッシュを停止するには、Ctrl-Cを押します。

```
# dlstat show-phys -r -i 1
LINK  TYPE  INDEX  IPKTS  RBYTES
net0   rx     0    101.91K  32.86M
net1   rx     0     9.61M  14.47G
net2   rx     8     336K    0
net0   rx     0      0      0
net1   rx     0     82.13K  123.69M
net2   rx     0      0      0
...
^C
```

この例は、ネットワークデバイスとしての net1 での送信リングの使用状況を示しています。

```
# dlstat show-phys -t net1
LINK  TYPE  INDEX  OPKTS  OBYTES
net1   tx     0      44    3.96K
net1   tx     1      0      0
net1   tx     2    1.48M  121.68M
net1   tx     3    2.45M  201.11M
net1   tx     4    1.47M  120.82M
net1   tx     5      0      0
net1   tx     6    1.97M  161.57M
net1   tx     7    4.59M  376.21M
```

```
net1 tx 8 2.43M 199.24M
net1 tx 9 0 0
net1 tx 10 3.23M 264.69M
net1 tx 11 1.88M 153.96M
```

## レーン上のネットワークトラフィックの統計情報の取得

dlstat show-link サブコマンドは、物理リンク上に構成されているレーンに関する統計情報を提供します。これらのレーンは、データリンクで構成されています。図 4-1 に示すように、このサブコマンドは、ネットワークスタックのデータリンク層で動作します。また、同等のサブコマンド dladm show-link も、このスタックの同じレベルで動作します。図 4-1 を、『Oracle Solaris 11 ネットワーキングの紹介』の「Oracle Solaris 11 の実装」に示されているネットワークスタックと比較してください。

次の例は、vnic0 の受信側のトラフィックの統計情報を示しています。

```
# dlstat show-link -r vnic0
LINK TYPE ID INDEX IPKTS RBYTES INTRS POLLS IDROPS
vnic0 rx hw 2 1.73M 2.61G 1.33M 400.22K 0
vnic0 rx hw 4 8.44M 12.71G 4.35M 4.09M 0
```

前の出力は、レーン vnic0 のトラフィックの統計情報を示しています。このレーンには、2つの受信リング(リング 2 とリング 4)が排他的使用のために割り当てられています。この出力は、これらの2つのリングが受信ネットワークトラフィックのためにどのように使用されているかを示しています。ただし、これらのデータには、帯域幅制限や優先処理などのほかのリソース割り当ての実装も反映されている可能性があります。

プライマリレーンである net0 に対して次の情報が表示されたとします。

```
# dlstat show-link -r net0
LINK TYPE ID INDEX IPKTS RBYTES INTRS POLLS IDROPS
net0 rx local -- 0 0 0 0 0
net0 rx sw -- 794.28K 1.19G 794.28K 0 0
...
```

この出力に基づくと、Rx リングの1つ(リング 0)が別のクライアントと共有されています。リングは、セカンダリクライアントが、リングの割り当てなしで構成されている場合に共有されます。リングが割り当てられない理由として、次のことが考えられます。

- ハードウェアクライアントをリンク上に作成することができなくなった。
- ハードウェアリングを割り当てることができなくなった。
- 管理者が意図的にソフトウェアクライアントを構成している。

割り込み (INTRS) と欠落 (\*DROPS) に関する統計情報も重要です。割り込みの数が少なく、パケットの欠落が 0 である場合は、パフォーマンスの効率が高いことを示しま

す。割り込みの数またはパケットの欠落数が多い場合は、このレーンにさらに多くのリソースを追加することが必要になる可能性があります。

次の例は、プライマリレーンである net1 によって使用されているリング上のアウトバウンドパケットに関する統計情報を示しています。この出力は、net1 がすべての Tx リングを使用していることを示しています。

```
# dlstat show-link -t net1
LINK  TYPE  ID  INDEX  OPKTS  OBYTES  ODROPS
net1   tx    hw   0      32     1.44K    0
net1   tx    hw   1       0       0        0
net1   tx    hw   2    1.48M   97.95M   0
net1   tx    hw   3    2.45M  161.87M   0
net1   tx    hw   4    1.47M   97.25M   0
net1   tx    hw   5       0      276      0
net1   tx    hw   6    1.97M  130.25M   0
net1   tx    hw   7    4.59M  302.80M   0
net1   tx    hw   8    2.43M  302.80M   0
net1   tx    hw   9       0       0        0
net1   tx    hw  10    3.23M  213.05M   0
net1   tx    hw  11    1.88M  123.93M   0
```

次のコマンドは、リンク net1 の受信側の統計情報の使用法を示しています。さらに、このコマンドでは -F オプションが使用されているため、出力にはファンアウト情報も表示されます。具体的には、ファンアウト数は2つ (0 と 1) です。リング 0 を使用するハードウェアレーン上で受信されたネットワークトラフィックは分割され、2つのファンアウトにわたって渡されます。同様に、リング 1 を使用するハードウェアレーン上で受信されたネットワークトラフィックも分割され、2つのファンアウトにわたって分配されます。

```
# dlstat show-link -r -F net1
LINK  ID  INDEX  FOUT  IPKTS
net1  local  --    0     0
net1  hw     0     0  382.47K
net1  hw     0     1     0
net1  hw     1     0  367.50K
net1  hw     1     1  433.24K
```

## リンクアグリゲーション上のネットワークトラフィックの統計情報の取得

dlstat show-aggr コマンドは、トラフィックがシステム上のアグリゲーションをたどるときの、アグリゲーションのポートごとのネットワークパケットの統計情報を表示します。

```
# dlstat show-aggr
LINK  PORT  IPKTS  RBYTES  OPKTS  OBYTES
aggr1  --    0      0       0      0
aggr1  net0  0      0       0      0
aggr1  net1  0      0       0      0
```

この出力は、ベースとなる2つのリンク `net0` と `net1` を含むリンクアグリゲーション `aggr1` の構成を示しています。ネットワークトラフィックがアグリゲーションを経由してシステムによって受信または送信されると、受信および送信パケットとそれらの各サイズに関する情報がポートごとに報告されます。これらのポートは、そのアグリゲーションのベースとなるリンクで識別されます。

## フロー上のネットワークトラフィックに関する統計情報の収集

フローの統計情報は、システム上で定義されている任意のフロー上のパケットトラフィックを評価するのに役立ちます。フロー情報を取得するには、`flowstat` コマンドを使用します。このコマンドについての詳細は、`flowstat(1M)` のマニュアルページを参照してください。

`flowstat` コマンドのもっとも一般的に使用される構文は次のとおりです。

```
# flowstat [-r|-t] [-i interval] [-l link] [flow]
```

- `[-r|-t]` 受信側の統計情報のみ (`-r` オプション) または送信側の統計情報のみ (`-t` オプション) のどちらかを表示します。受信側と送信側の両方に関する統計情報を表示するには、このオプションを省略します。
- `-i interval` 表示されている統計情報がリフレッシュされる間隔 (秒単位) を指定します。このオプションを使用しない場合は、静的な出力が表示されません。
- `-l link` 指定されたデータリンク上のすべてのフローの統計情報を監視することを示します。このオプションを使用しない場合は、すべてのデータリンク上のすべてのフローに関する情報が表示されます。
- `flow` 指定されたフローのみの統計情報を監視することを示します。このオプションを使用しない場合は、リンクを指定したかどうかに応じて、すべてのフローの統計情報が表示されます。

次の例は、システム上で構成されているフローに関する情報を表示するためのさまざまな方法を示しています。

例 4-1 すべてのフローのトラフィック統計情報の1秒間隔での表示

この例では、システム上で構成されているすべてのフロー上の受信および送信トラフィックに関する情報を1秒ごとに表示します。

```
# flowstat -i 1
FLOW  IPKTS  RBYTES  IERRS  OPKTS  OBYTES  OERRS
flow1  528.45K  787.39M  0      179.39K  11.85M  0
flow2  742.81K  1.10G   0      0        0        0
flow3  0        0       0      0        0        0
```

例 4-1 すべてのフローのトラフィック統計情報の 1 秒間隔での表示 (続き)

```

flow1 67.73K 101.02M 0 21.04K 1.39M 0
flow2 0 0 0 0 0 0
flow3 0 0 0 0 0 0
...
^C

```

この例は、構成されているすべてのフローの送信トラフィックに関する統計情報を示しています。

例 4-2 すべてのフローに関する送信側の統計情報の表示

```

# flowstat -t
FLOW  OPKTS  OBYTES  OERRS
flow1 24.37M  1.61G   0
flow2 0        0       0
flow1 4        216     0

```

例 4-3 指定されたリンク上のすべてのフローの受信側の統計情報の表示

この例は、データリンク net0 上に作成されたすべてのフロー上のハードウェアレーン内の受信トラフィックを示しています。

```

# flowstat -r -i 2 -l net0
FLOW  IPKTS  RBYTES  IERRS
tcp-flow 183.11K 270.24M 0
udp-flow 0 0 0
tcp-flow 373.83K 551.52M 0
udp-flow 0 0 0
tcp-flow 372.35K 549.04M 0
udp-flow 0 0 0
tcp-flow 372.87K 549.61M 0
udp-flow 0 0 0
tcp-flow 371.57K 547.89M 0
udp-flow 0 0 0
tcp-flow 191.92K 282.95M 0
udp-flow 206.51K 310.70M 0
tcp-flow 0 0 0
udp-flow 222.75K 335.15M 0
tcp-flow 0 0 0
udp-flow 223.00K 335.52M 0
tcp-flow 0 0 0
udp-flow 160.22K 241.07M 0
tcp-flow 0 0 0
udp-flow 167.89K 252.61M 0
tcp-flow 0 0 0
udp-flow 9.52K 14.32M 0
^C

```

# ネットワークトラフィックのためのネットワークアカウンティングの構成

拡張アカウンティング機能を使用すると、システム上のネットワークアカウンティングを設定できます。ネットワークアカウンティングでは、ネットワークトラフィックに関する統計情報をログファイルに取得します。この方法では、追跡、プロビジョニング、統合、および請求の目的でトラフィックのレコードを保持することができます。あとで、一定期間にわたるネットワーク使用に関する履歴情報を取得するために、このログファイルを参照できます。

ネットワークアカウンティングを設定するには、拡張アカウンティング機能の `acctadm` コマンドを使用します。ネットワークアカウンティングの設定を完了したら、`flowstat` コマンドを使用してトラフィックの統計情報を記録します。

このセクションでは、次の手順について説明します。

- 67 ページの「ネットワークアカウンティングを設定する方法」
- 69 ページの「ネットワークトラフィックに関する履歴統計情報を取得する方法」

## ▼ ネットワークアカウンティングを設定する方法

- 1 ネットワーク使用を追跡するインタフェースを備えたシステムで、管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 拡張アカウンティング機能で有効にすることのできるアカウンティングタイプのステータスを表示します。

```
# acctadm [process | task | flow | net]
```

拡張アカウンティング機能では、4つのタイプのアカウンティングを有効にすることができます。`acctadm` コマンドのオプションのオペランドは、これらのアカウンティングタイプに対応しています。このコマンドのオペランドを使用して、特定のタイプのアカウンティングを構成します。

- プロセスアカウンティング
- タスクアカウンティング
- IPQoS のフローアカウンティング
- リンクとフローのネットワークアカウンティング

---

注- ネットワークアカウントティングはまた、50ページの「フロー上のリソースの管理」で説明されている `flowadm` および `flowstat` コマンドで管理されるフローにも適用されます。そのため、これらのフローのためのアカウントティングを設定するには、`acctadm` コマンドで `net` オプションを使用します。IPQoS 構成のためのフローアカウントティングを有効にする `flow` オプションを使用しないでください。

`net` を指定すると、ネットワークアカウントティングのステータスが表示されます。`net` が使用されない場合は、4つのすべてのアカウントティングタイプのステータスが表示されます。

---

- 3 ネットワークトラフィックのための拡張アカウントティングを有効にします。

```
# acctadm -e extended -f filename net
```

ここで、`filename` には、ネットワークトラフィックの統計情報を取得するログファイルのフルパスが含まれます。このログファイルは、指定した任意のディレクトリ内に作成できます。

- 4 拡張ネットワークアカウントティングがアクティブになっていることを確認します。

```
# acctadm net
```

#### 例 4-4 システム上でのネットワークアカウントティングの設定

この例は、システム上の履歴トラフィック情報を取得して表示するようにネットワークアカウントティングを構成する方法を示しています。

最初に、次のように、すべてのアカウントティングタイプのステータスを表示します。

```
# acctadm
Task accounting: inactive
Task accounting file: none
Tracked task resources: none
Untracked task resources: extended
Process accounting: inactive
Process accounting file: none
Tracked process resources: none
Untracked process resources: extended,host
Flow accounting: inactive
Flow accounting file: none
Tracked flow resources: none
Untracked flow resources: extended
Network accounting: inactive
Network accounting file: none
Tracked Network resources: none
Untracked Network resources: extended
```

この出力は、ネットワークアカウントिंगがアクティブになっていないことを示します。

次に、拡張ネットワークアカウントिंगを有効にします。

```
# acctadm -e extended -f /var/log/net.log net
# acctadm net
    Net accounting: active
    Net accounting file: /var/log/net.log
    Tracked net resources: extended
    Untracked net resources: none
```

## ▼ ネットワークトラフィックに関する履歴統計情報を取得する方法

ネットワークアカウントिंगを有効にしたら、`dlstat` および `flowstat` コマンドを使用して、ログファイルから情報を抽出できます。ここでは、その手順について説明します。

始める前に ネットワークに関する履歴データを表示するには、その前にネットワークのための拡張アカウントINGを有効にする必要があります。さらに、フロー上のトラフィックに関する履歴データを表示するには、まず [50 ページの「フロー上のリソースの管理」](#) の説明に従って、システム上のフローを構成する必要があります。

- 1 ネットワーク使用を追跡するインタフェースを備えたシステムで、管理者になります。  
詳細は、『[Oracle Solaris 11.1 の管理: セキュリティサービス](#)』の「[割り当てられている管理権限を使用する方法](#)」を参照してください。
- 2 データリンク上のリソース使用状況に関する履歴情報を抽出して表示するには、次のコマンドを使用します。

```
# dlstat show-link -h [-a] -f filename [-d date] [-F format] [-s start-time] [-e end-time] [link]
-h                リソース使用状況に関する履歴情報のサマリーをデータリンク上の受信および送信パケットごとに表示します。
-a                すべてのデータリンク (データ取得のあとにすでに削除されたデータリンクを含む) 上のリソース使用状況を表示します。
-f filename       acctadm コマンドでネットワークアカウントINGが有効にされたときに定義されたログファイルを指定します。
-d date           指定された日付に対するログ記録された情報を表示します。
```

- F *format*           あとで解析のためにプロットできる特定の形式でデータを表示します。現在、サポートされている形式は `gnuplot` だけです。
  - s *start-time*,  
-e *end-time*           指定された日付と時間範囲に対する使用可能なログ記録された情報を表示します。MM/DD/YYYY, hh:mm:ss の形式を使用します。hour (hh) には、24時間制のクロック表記を使用する必要があります。日付を含めない場合は、現在の日付の指定された時間範囲のデータが表示されます。
  - link*                 指定されたデータリンクの履歴データを表示します。このオプションを使用しない場合は、構成されているすべてのデータリンクの履歴ネットワークデータが表示されます。
- 3** 構成されているフロー上のネットワークトラフィックに関する履歴情報を抽出して表示するには、次のコマンドを使用します。
- ```
# flowstat -h [-a] -f filename [-d date] [-F format] [-s start-time] [-e end-time] [flow]
```
- h                   リソース使用状況に関する履歴情報のサマリーを、構成されているフロー上の受信および送信パケットごとに表示します。
  - a                   構成されているすべてのフロー(データ取得のあとにすでに削除されたフローを含む)上のリソース使用状況を表示します。
  - f *filename*         acctadm コマンドでネットワークアカウントティングが有効にされたときに定義されたログファイルを指定します。
  - d                   指定された日付に対するログ記録された情報を表示します。
  - F *format*           データを特定の形式で表示します。現在、サポートされている形式は `gnuplot` だけです。
  - s *start-time*,  
-e *end-time*           指定された日付と時間範囲に対する使用可能なログ記録された情報を表示します。MM/DD/YYYY, hh:mm:ss の形式を使用します。hour (hh) には、24時間制のクロック表記を使用する必要があります。日付を含めない場合は、現在の日付の指定された時間範囲のデータが表示されます。
  - flow*                指定されたフローの履歴データを表示します。このオプションを使用しない場合は、構成されているすべてのフローの履歴ネットワークデータが表示されます。

#### 例 4-5 データリンク上のリソース使用状況に関する履歴情報の表示

次の例は、指定されたデータリンク上のネットワークトラフィックとそのリソースの使用に関する履歴統計情報を示しています。

```
# dlstat show-link -h -f /var/log/net.log net0
LINK DURATION IPACKETS RBYTES OPACKETS OBYTES BANDWIDTH
net0 80 1031 546908 0 0 2.44 Kbps
```

#### 例 4-6 フロー上のリソース使用状況に関する履歴情報の表示

次の例は、フロー上のネットワークトラフィックとそのリソースの使用に関する履歴統計情報を表示するための各種の方法を示しています。

次の例では、リソース使用状況の履歴統計情報をフロー上のトラフィックごとに表示します。

```
# flowstat -h -f /var/log/net.log
FLOW DURATION IPACKETS RBYTES OPACKETS OBYTES BANDWIDTH
flowtcp 100 1031 546908 0 0 43.76Kbps
flowudp 0 0 0 0 0 0.00Mbps
```

次の例では、特定の日付と時間範囲にわたるリソース使用状況の履歴統計情報をフロー上のトラフィックごとに表示します。

```
# flowstat -h -s 02/19/2008,10:39:06 -e 02/19/2008,10:40:06 \
-f /var/log/net.log flowtcp
FLOW START END RBYTES OBYTES BANDWIDTH
flowtcp 10:39:06 10:39:26 1546 6539 3.23 Kbps
flowtcp 10:39:26 10:39:46 3586 9922 5.40 Kbps
flowtcp 10:39:46 10:40:06 240 216 182.40 bps
flowtcp 10:40:06 10:40:26 0 0 0.00 bps
```

次の例では、特定の日付と時間範囲にわたるリソース使用状況の履歴統計情報をフロー上のトラフィックごとに表示します。これらの情報は、gnuplot形式を使用して表示されます。

```
# flowstat -h -s 02/19/2008,10:39:06 -e 02/19/2008,10:40:06 \
-F gnuplot -f /var/log/net.log flowtcp
# Time tcp-flow
10:39:06 3.23
10:39:26 5.40
10:39:46 0.18
10:40:06 0.00
```



# 索引

---

## A

acctadm コマンド, 67-71

## C

CPU, 15, 45

    CPU の割り当て, 49-50

CPU プール, 15, 45

    デフォルトプール, 46

    リンクへの割り当て, 47

## D

dladm コマンド

    create-etherstub, 12, 18

    create-vlan, 36

    create-vnic, 11-12, 14, 18

    delete-vnic, 32-33

    modify-vnic, 29

    set-linkprop, 13, 14

    show-linkprop, 38

    show-phys, 38

    show-vnic, 32

dlstat コマンド, 57, 60

    show-aggr, 64-65

    show-link, 63-64

    show-phys, 61-63

DS フィールド, 14

## E

etherstub, 8-11, 17-19

    作成, 12, 18

    プライベート仮想ネットワーク, 25-28

## F

flowadm コマンド, 50-55

    add-flow, 15, 51

    set-flowprop, 15, 51

    show-flowprop, 52

    ヘルプ, 15

flowstat コマンド, 57, 60, 65-66

## G

GARP VLAN 登録プロトコル (GVRP), 19

## I

ipadm コマンド

    create-addr, 18

    create-ip, 18

    IP 情報の表示, 54

## M

MAC アドレス, VNIC 上, 8

MAC クライアント, 35

## MACクライアント(続き)

- 構成, 43-44
- セカンダリ, 36, 42
- ソフトウェアベース, 35
- ハードウェアベース, 35
- プライマリ, 36, 41
- リングの割り当て, 43-44

## N

- NICリング, 14
  - 「リングのグループ化」も参照

## S

- SCTP, 14

## T

- TCP, 14

## U

- UDP, 14

## V

- VLAN
  - VNICとして, 19-20
  - VNICのVLANIDの変更, 29
- VNIC, 7-12
  - CPUプールリソースの割り当て, 47
  - MACアドレスの変更, 29-30
  - VLANID, 11
  - VLANIDの変更, 29
  - VLANIDを持つ, 19-20
  - VNICのIPインタフェースの作成, 18
  - 移行, 30-32
  - 構成, 21-23
  - 削除, 32-33

## VNIC(続き)

- 作成, 11-12, 18
- 情報の表示, 32
- ゾーンでの使用, 23-25
- ゾーンへの割り当て, 24
- 帯域幅, 14
- プライベート仮想ネットワーク, 25-28
- プロパティの設定, 14
- ベースとなるリンクの変更, 30-32
- VNICの移行, 30-32

## い

- インタフェース専用のCPU, 49-50

## か

- 外部仮想ネットワーク, 7-12
- 拡張アカウンティング機能, 67-71
  - IPQoSのフローアカウンティング, 67
  - タスクアカウンティング, 67
  - プロセスアカウンティング, 67
  - リンクとフローのネットワークアカウンティング, 67
- 仮想化とサービスの品質, 35
- 仮想スイッチ, 8-11
- 仮想ネットワーク
  - 「VNIC」も参照
  - etherstub, 17-19
  - VNICの構成, 17-19
  - 外部および内部, 7-12
  - 構築, 20-28
  - ゾーンでの使用, 9
  - ゾーンの構成, 21-23, 23-25
  - 排他的なIPタイプのゾーン, 24
  - プライベート, 7-12
- 仮想ネットワークカード, 「VNIC」を参照

## き

- 擬似EthernetNIC, 「etherstub」を参照

## さ

サービスの品質 (QoS), 12-13  
サービスレベル契約 (SLA), 13

## せ

静的なリングのグループ化, 35-44  
セカンダリクライアント, 36, 42

## そ

## ゾーン

VNICの割り当て, 24  
仮想ネットワークのための構成, 23-25  
ネットワーク仮想化のための構成, 21-23  
属性, フロー, 13  
ソフトウェアベースのクライアント, 35

## た

## 帯域幅

VNIC, 14  
フロー上, 15  
フロー上の設定, 51

## て

データセンターの統合, 11  
データリンク  
リソース制御プロパティ, 12-13, 14  
データリンク, リソース制御プロパティ, 14

## と

動的なリングのグループ化, 35-44  
トランスポートプロトコル, 14

## な

内部仮想ネットワーク, 7-12

## ね

ネットワークアカウントिंग, 67-71  
ネットワーク仮想化, 7-12  
etherstub, 8-11  
VNIC, 8-11  
仮想スイッチ, 8-11  
コンポーネント, 8-11  
実装のための dladm サブコマンド, 14  
データセンターの統合, 11  
ネットワーク使用の監視, 57  
ネットワークの統計情報, 60-65  
アグリゲーションポート上のトラフィック, 64-65  
ネットワーク使用の監視, 57  
ネットワークトラフィックのためのデバイスの使用, 61-63  
フロー上のリソース使用状況, 65-66  
履歴トラフィック情報, 68, 69-71  
レーン上のネットワークトラフィック, 63-64  
ネットワークリソースの管理, 35  
CPU, 45  
CPU プール, 45  
実装のための dladm サブコマンド, 14  
フローの使用, 13  
リンク上, 12  
ネットワークレーン, 12, 13  
ソフトウェアレーン, 57  
ハードウェアレーン, 57

## は

ハードウェアベースのクライアント, 35  
ハードウェアリング, 35-44  
パケットの欠落, 63

## ふ

プライベート仮想ネットワーク, 7-12

プライベート仮想ネットワーク (続き)

  etherstub を使用した構成, 25-28

プライマリクライアント, 36, 41

フロー, 13, 50-55

  作成, 15, 51

  情報の表示, 52

  属性, 13

  帯域幅の設定, 15, 51

  プロパティの設定, 15

フロー制御, 50-55

## り

リソース制御, 12-13, 35

リング, 送信および受信, 35-44

リングのグループ化

  NICでのサポート, 38-41

  VLAN上, 36

  受信および送信リング, 38

  詳細情報の表示, 37-38

  データリンクプロパティ, 37-38

  動的および静的, 35-44

  リング情報の解釈, 39, 41

  リングの使用とリングの割り当て, 41-43

  割り当ての手順, 43-44

リング割り当て, 「リングのグループ化」を参照

## わ

割り込み, 50, 63