Oracle® Solaris 11.2 での仮想ネットワークとネットワークリソースの管理

Part No: E53791-02
2014 年 9 月

ORACLE®

Copyright © 2011, 2014, Oracle and/or its affiliates. All rights reserved.

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に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

このドキュメントの使用法 9

1 ネットワーク仮想化およびネットワークリソースの管理の概要 11
Oracle Solaris 11.2 のネットワーク仮想化およびネットワークリソースの管理の新機能 11
ネットワーク仮想化およびネットワークリソースの管理とは 13
仮想ネットワークの概要 14
仮想ネットワークのコンポーネント 15
仮想ネットワークの仕組み 18
仮想拡張ローカルエリアネットワークの使用 20
エッジ仮想ブリッジングの使用 21
仮想ネットワークを実装すべきユーザー 21
ネットワークリソースの管理の概要 21
データリンクプロパティーを使用することによるネットワークリソースの管理 22
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 23
ネットワークリソースの管理の利点 23

2 仮想ネットワークの作成および管理 25
仮想ネットワークのコンポーネントの構成 25
仮想ネットワークのコンポーネントを構成するためのコマンド 25
▼ VNIC と etherstub を構成する方法 27
▼ VLAN ID を持つ VNIC を構成する方法 28
仮想ネットワークの構築 30
▼ 仮想ネットワークのためにゾーンを構成する方法 31
▼ ゾーンを VNIC を使用するように再構成する方法 33
▼ ゾーンで一時的に VNIC を作成する方法 36
▼ プライベート仮想ネットワークを構成する方法 37
VNIC の管理 40
VNIC の表示 41
VNIC の VLAN ID の変更 45
VNIC の MAC アドレスの変更 46

VNIC の移行 48
VNIC の削除 50
VNIC でのシングルルート I/O 仮想化の使用 53
データリンクの SR-IOV モードの有効化 53
VF VNIC の作成 54
VF VNIC の管理 56
VF 情報の表示 57

3 仮想拡張ローカルエリアネットワークを使用することによる仮想ネットワークの構成 59
VXLAN の概要 60
VXLAN を使用することの利点 60
VXLAN の命名規則 61
VXLAN トポロジ 62
ゾーンとの VXLAN の使用 64
VXLAN 構成の計画 66
VXLAN の要件 66
VXLAN の構成 67
▼ VXLAN を構成する方法 67
VXLAN 情報の表示 72
VXLAN の削除 72
ゾーンへの VXLAN の割当て 72
▼ ゾーンに VXLAN を割り当てる方法 73
ユースケース: リンクアグリゲーション上に VXLAN の構成 74

4 エッジ仮想ブリッジングを使用したサーバーネットワークエッジの仮想化の管理 79
サーバーネットワークエッジの仮想化における EVB のサポート 80
反射型リレー 80
ネットワーク内の VNIC の自動構成 81
Improving Network and Server Efficiency by Using EVB 81
EVB のインストール 84
▼ EVB をインストールする方法 84
同じ物理ポート上の VM 間の切り替えの制御 85
外部スイッチ経由での VM の通信の有効化 85
VM 間の通信を管理するための LLDP の使用 88
VDP を使用した VNIC 情報の交換 90
VDP による VNIC 情報の交換方法 91
VDP と ECP の状態と統計情報 92
VDP の状態と統計情報の表示 92
リンクプロパティーの表示 93

ECP の状態と統計情報の表示 94
デフォルトの EVB 構成の変更 94
▼ デフォルトの EVB 構成を変更する方法 96

5 エラスティック仮想スイッチについて 99
エラスティック仮想スイッチ (EVS) 機能の概要 99
Oracle Solaris での仮想スイッチ 100
Oracle Solaris のエラスティック仮想スイッチ機能とは 101
EVS を使用する利点 103
エラスティック仮想スイッチのリソース 104
EVS での名前空間の管理 106
EVS のコンポーネント 106
EVS マネージャー 107
EVS コントローラ 108
EVS クライアント 110
EVS ノード 111
EVS 管理コマンド 111
`evsadm` コマンド 111
`evsstat` コマンド 113
`dladm` コマンド 114
`zonecfg` コマンド 114
エラスティック仮想スイッチに接続された VNIC を管理する場合の制限 115
自動的に生成される VXLAN データリンク 115
EVS を使用するための必須パッケージ 116
ゾーンでの EVS の機能 117
EVS を使用するためのセキュリティー要件 117

6 エラスティック仮想スイッチの管理 119
EVS 管理タスク 119
エラスティック仮想スイッチ構成の計画 120
EVS コントローラの作成と管理 121
EVS コントローラの必須パッケージ 122
EVS コントローラを構成するためのコマンド 122
EVS コントローラの構成 125
エラスティック仮想スイッチの構成 132
エラスティック仮想スイッチの必須パッケージ 133
エラスティック仮想スイッチを構成するためのコマンド 133
▼ エラスティック仮想スイッチを構成する方法 135
エラスティック仮想スイッチ用の VNIC の作成 137

エラスティック仮想スイッチ、IPnet、および VPort の管理 139
エラスティック仮想スイッチの管理 139
IPnet 構成の管理 144
VPort 構成の管理 145
エラスティック仮想スイッチの削除 150
エラスティック仮想スイッチのモニタリング 151
エラスティック仮想スイッチのユースケースの例 154
ユースケース: エラスティック仮想スイッチの構成 154
ユースケース: テナント用のエラスティック仮想スイッチの構成 161

7 ネットワークリソースの管理 169
データリンクプロパティーを使用することによるネットワークリソースの管理 169
データリンクにリソースを割り当てるためのコマンド 170
NIC リングの管理 171
MAC クライアントへのリングの割り当て 171
VLAN でのリングの割り当て 171
リングを構成するためのコマンド 172
データリンク上でのリングの使用とリング割り当ての表示 174
クライアントの構成とリングの割り当て 175
プールと CPU の管理 181
プールおよび CPU の操作 182
データリンク用の CPU プールの構成 184
データリンクへの CPU の割り当て 185
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 186
フローへのリソース割り当てのためのコマンド 188
フローの構成 188
ユースケース: データリンクとフローのプロパティーを設定することによるネットワークリソースの管理 190

8 ネットワークトラフィックとリソース使用状況のモニタリング 197
データリンクとフローのネットワークトラフィックの統計情報のモニタリングの概要 197
ネットワークトラフィックの統計情報をモニターするためのコマンド 200
リンクのネットワークトラフィックの統計情報の表示 200
ネットワークデバイスのネットワークトラフィックの統計情報の表示 201
データリンクのネットワークトラフィックの統計情報の表示 204
リンクアグリゲーションのネットワークトラフィックの統計情報の表示 205
ブリッジのネットワークトラフィックの統計情報の表示 206
フロー上のネットワークトラフィックの統計情報の表示 206
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 209

▼ ネットワークアカウンティングを設定する方法 209
ネットワークトラフィックに関する履歴統計情報の表示 211

索引 215

このドキュメントの使用法

- **概要** – Oracle Solaris の仮想ネットワーク機能を構成し、ネットワークトラフィックをモニターする方法について説明します。また、ネットワークリソースを管理するために使用されるさまざまなプロセスについても説明します。
- **対象読者** – システム管理者。
- **前提知識** – ネットワーク管理の基本的なスキルと一部の高度なスキル。

製品ドキュメントライブラリ

この製品の最新情報や既知の問題は、ドキュメントライブラリ (http://www.oracle.com/pls/topic/lookup?ctx=E36784) に含まれています。

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 (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。

フィードバック

このドキュメントに関するフィードバックを http://www.oracle.com/goto/docfeedback からお聞かせください。

♦ ♦ ♦ 第 1 章

ネットワーク仮想化およびネットワークリソースの管理の概要

この章では、Oracle Solaris でのネットワーク仮想化とネットワークリソースの管理の概要を説明します。

この章の内容は、次のとおりです。

- 11 ページの「Oracle Solaris 11.2 のネットワーク仮想化およびネットワークリソースの管理の新機能」
- 13 ページの「ネットワーク仮想化およびネットワークリソースの管理とは」
- 14 ページの「仮想ネットワークの概要」
- 21 ページの「ネットワークリソースの管理の概要」

Oracle Solaris 11.2 のネットワーク仮想化およびネットワークリソースの管理の新機能

既存のユーザーのため、このセクションでは、このリリースでの重要な変更点を示します。

- **Oracle Solaris エラスティック仮想スイッチ (EVS) 機能** - Oracle Solaris ネットワーク仮想化機能は、仮想スイッチを直接管理できるように拡張されています。Oracle Solaris エラスティック仮想スイッチ機能は、複数のサーバー上に存在する仮想マシンを相互接続するデータセンターやマルチテナントクラウド環境に、仮想ネットワークインフラストラクチャーを提供します。同じエラスティック仮想スイッチに接続された仮想マシンは、互いに通信できます。EVS を使用すると、複数のホスト上の仮想スイッチを集中管理でき、したがってエラスティック仮想スイッチに接続された VNIC を集中管理できます。詳細は、第5章「エラスティック仮想スイッチについて」を参照してください。エラスティック仮想スイッチを管理する方法についての詳細は、第6章「エラスティック仮想スイッチの管理」を参照してください。
- **仮想拡張ローカルエリアネットワーク (VXLAN) のサポート** - Oracle Solaris は、大規模なデータセンターでの仮想化をサポートするための分離方法を提供する、VXLAN テクノロジ

をサポートします。これにより、クラウド環境で異なるレイヤー 2 ネットワークに属する物理サーバーの間で、仮想マシンを移行できます。詳細は、第3章「仮想拡張ローカルエリアネットワークを使用することによる仮想ネットワークの構成」を参照してください。

- **シングルルート I/O 仮想化 (SR-IOV) のサポート** - この機能により、SR-IOV をサポートするネットワークデバイスに、仮想機能 (VF) ベースの VNIC を作成できます。詳細は、53 ページの「VNIC でのシングルルート I/O 仮想化の使用」を参照してください。
- **ゾーンでの仮想ネットワークインタフェースカード (VNIC) の一時的な作成** – VNIC を一時的に、大域ゾーンから非大域ゾーン内に直接作成できます。一時的な VNIC を作成するには、`dladm create-vnic` コマンドで `-t` オプションを使用する必要があります。一時的な VNIC はゾーンの次回のリブートまで保持されます。一時的な VNIC の作成に加えて、ゾーンに VLAN と IP over InfiniBand (IPoIB) のパーティションを一時的に作成することもできます。詳細は、36 ページの「ゾーンで一時的に VNIC を作成する方法」を参照してください。
- **外部スイッチを使用することによる VNIC 間の通信** - Oracle Solaris 11.2 の反射型リレー機能を使用すると、ベースとなる同じ物理 NIC を共有するローカルの Oracle Solaris ゾーンまたは Oracle VM の間のトラフィックが、ホストの仮想スイッチではなく、常に強制的に物理ネットワークに送信されるようにすることができます。これらのエンティティー間の通信は、反射型リレー機能をサポートする外部スイッチ上で構成されたポリシーに従います。詳細は、85 ページの「同じ物理ポート上の VM 間の切り替えの制御」を参照してください。
- **ネットワークトラフィックの統計情報をモニターするための拡張機能** - 拡張された `dlstat` コマンドと `flowstat` コマンドを使用すると、ネットワークトラフィックの統計情報を効果的にモニターできます。ネットワークトラフィックの統計情報をモニターするための拡張機能は次のとおりです。
  - ネットワークトラフィックの統計情報は現在の時間とともに表示されます。
  - ネットワークトラフィックの統計情報は、指定された間隔とカウントの値に基づいて表示およびリフレッシュされます。
  - ネットワークトラフィックの統計情報は、指定された間隔値に基づく 1 秒あたりの変化量で表示されます。

  拡張機能の詳細は、201 ページの「ネットワークデバイスのネットワークトラフィックの統計情報の表示」および 206 ページの「フロー上のネットワークトラフィックの統計情報の表示」を参照してください。
- **フロー構成の変更** – 拡張された `flowadm add-flow` コマンドを使用すると、さらに多くの属性とその新しい組み合わせに基づいてデータリンク上にフローを構成することができ、さまざまなポート、トランスポートプロトコル、および IP アドレスから受信されるネットワークパケット

を選択的に整理できます。詳細は、186 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。

フローを管理するために使用する帯域幅プロパティーに加えて、`flowadm set-flowprop` コマンドを使用してフローの `priority` プロパティーを設定できます。`priority` プロパティーを設定することによって、フローの優先順位を設定できます。新しい読み取り専用プロパティー、`hwflow` を使用すると、フローがどのようにインスタンス化されているかを確認できます。詳細は、188 ページの「フローの構成」を参照してください。

- **VNIC に関連付けられた複数の MAC アドレスの表示** - 拡張された `dladm show-vnic` コマンドを使用すると、VNIC に関連付けられた複数の MAC アドレスを表示できます。詳細は、42 ページの「複数の MAC アドレスを持つ VNIC を表示」を参照してください。
- **システムで作成された VNIC** - `dladm create-vnic` コマンドを使用することによって作成できる VNIC に加えて、システムも VNIC を作成します (システムで作成された VNIC と呼ばれます)。詳細は、25 ページの「仮想ネットワークのコンポーネントを構成するためのコマンド」を参照してください。
- **データリンクの物理および仮想リンク状態の表示** – `dladm show-phys` コマンドと `dladm show-ether` コマンドを使用すると、データリンクの物理リンク状態を表示できます。データリンクの仮想リンク状態を表示するには、`dladm show-link` コマンドを使用できます。詳細は、43 ページの「データリンクの物理および仮想リンク状態の表示」を参照してください。
- **データリンクプロパティーの有効値の表示** - `dladm show-linkprop` コマンドは、データリンクプロパティーの `EFFECTIVE` フィールドを表示するように拡張されています。`EFFECTIVE` フィールドの値は、リソースの可用性、ベースとなるデバイスの容量、またはピアとのネゴシエーションに基づいて、システムによって決定されます。有効値は構成した値と同じである必要はありません。データリンクプロパティーは、プロパティーに値が構成されていない場合でも、有効値を持つことができます。

ネットワーク仮想化およびネットワークリソースの管理とは

*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。この 1 つの管理単位を、仮想ネットワークと呼びます。

*ネットワークリソースの管理*は、ネットワークプロセスのためのリソースの管理と割り当てのプロセスです。処理するネットワークトラフィックの量に応じて、異なる方法でリソースを割り当てることが

できます。実際のニーズに応じてリソースを管理および割り当てることによって、パケットを処理する際のシステムの効率が向上します。

ネットワーク仮想化は、ネットワークリソースの管理とともにを効果的に使用する場合に合わせて最適化されています。システムとユーザーがハードウェアとソフトウェアのネットワークリソースの共有を制御できるようにするため、仮想ネットワークプロセスの効率が向上します。ネットワーク仮想化とネットワークリソースの管理は、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、オペレーティングシステム (OS) 仮想化、ユーティリティー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。

仮想ネットワークの概要

*仮想ネットワーク*とは、物理ネットワークをエミュレートするネットワークであり、ハードウェアとソフトウェアのネットワークリソースの組み合わせです。仮想ネットワークは、ネットワーク仮想化の最終成果物です。

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。VLAN についての詳細は、『Oracle Solaris 11.2 でのネットワークデータリンクの管理 』の第 3 章「仮想ローカルエリアネットワークを使用した仮想ネットワークの構成」を参照してください。

*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 枚の物理ネットワークインタフェースカード (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。これらの仮想マシンまたはゾーンは、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。このドキュメントの各章では、内部仮想ネットワークに焦点を置いています。

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 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、外部ネットワーク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。*etherstub* と呼ばれる擬似 NIC 上に VNIC を構成することに

よって、この内部ネットワークをほかの外部ネットワークから切り離すことができます。詳細は、16 ページの「etherstub」を参照してください。

仮想ネットワークのコンポーネント

仮想ネットワークには次のコンポーネントがあります。

- 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
- ゾーン

仮想ネットワークインタフェースカード (VNIC)

VNIC は、構成されている場合は物理 NIC であるかのように動作する L2 エンティティー、つまり仮想ネットワークデバイスです。ベースとなるデータリンク上に VNIC を構成して、複数のゾーンまたは VM 間で共有します。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。すべての物理 Ethernet インタフェースが VNIC の作成をサポートします。VNIC を構成する方法についての詳細は、27 ページの「VNIC と etherstub を構成する方法」を参照してください。

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、自動的に生成される MAC アドレス以外の MAC アドレスを VNIC に割り当てることができます。詳細は、46 ページの「VNIC の MAC アドレスの変更」を参照してください。

仮想スイッチ

*仮想スイッチ*は、仮想マシン (VM) 間での通信を容易にするエンティティーです。仮想スイッチは物理マシン内の仮想マシン間のトラフィック (VM 間トラフィック) をループし、このトラフィックをネットワーク上では送信しません。仮想スイッチは、ベースとなるデータリンクの最上位に VNIC を作成するたびに、暗黙的に作成されます。VM が構成されている VNIC は、VM 間通信のためには同じ VLAN または VXLAN 上に存在する必要があります。仮想スイッチは EVS で管理できます。EVS の詳細は、第5章「エラスティック仮想スイッチについて」を参照してください。

Ethernet 設計に従って、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 Ethernet 設計は、仮想ネットワークが構成されたシステムでは制限となります。この Ethernet 設計の制限は仮想スイッチを使用することによって解決され、それにより VM が互いに通信できるようになります。

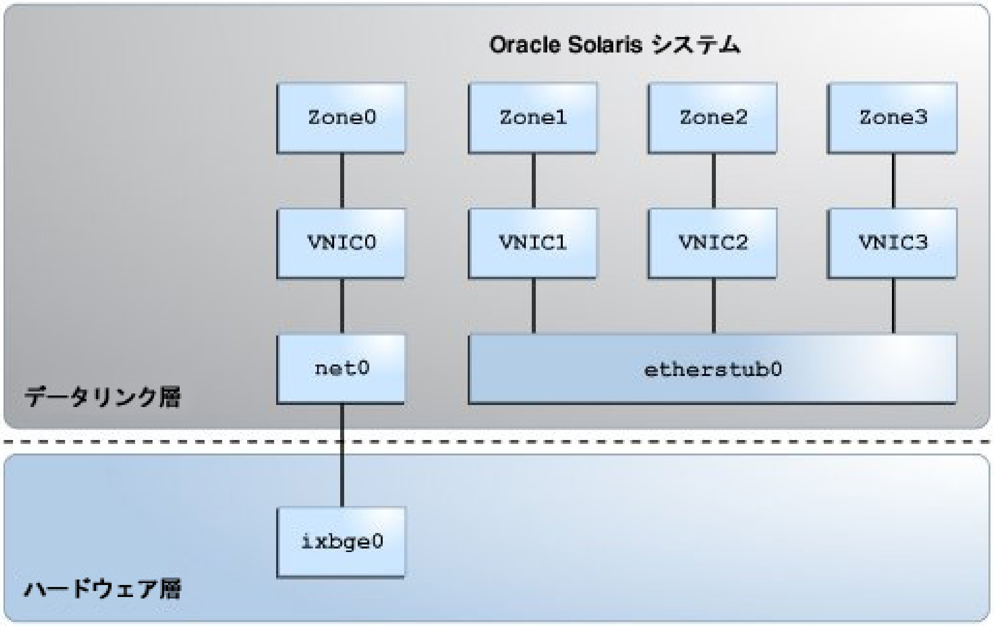
場合によっては、システムの VM 間の通信にスイッチの使用が必要になる場合があります。たとえば、スイッチに構成されているアクセス制御リスト (ACL) に従って VM 間の通信を行う必要がある場合などです。デフォルトでは、スイッチは、パケットを受信した同じポートでパケットを送信することはできません。したがって、スイッチを使用する VM 間の通信には、スイッチでの反射型リレーが有効になります。反射型リレーにより、スイッチは、パケットを受信した同じポートでパケットを転送できます。詳細は、反射型リレーを参照してください。

etherstub

etherstub は、Oracle Solaris ネットワークスタックのデータリンク層 (L2) に構成された擬似 Ethernet NIC です。VNIC は物理 NIC 上にではなく、etherstub 上に作成できます。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、etherstub を使用すると、アクセスがネットワーク全体にではなく、開発者のみに制限されたネットワーク環境を作成できます。

次の図は、etherstub に基づいたプライベート仮想ネットワークを示しています。

図 1-1 プライベート仮想ネットワーク



この図は、`VNIC1`、`VNIC2`、および `VNIC3` が構成されている `etherstub0` を示しています。各 VNIC が 1 つのゾーンに割り当てられます。etherstub に基づくプライベート仮想ネットワークには、外部ネットワークからはアクセスできません。詳細は、37 ページの「プライベート仮想ネットワークを構成する方法」を参照してください。

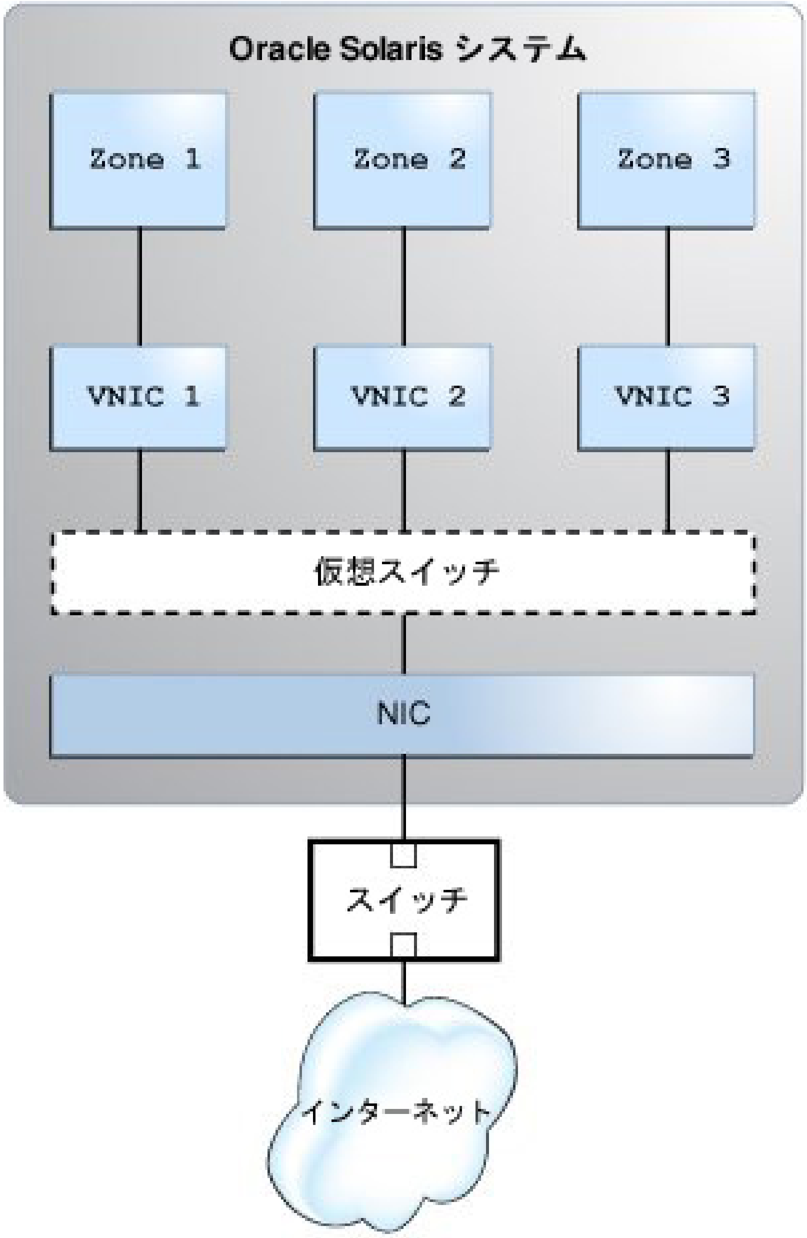
ゾーン

ゾーンは、Oracle Solaris オペレーティングシステムの 1 つのインスタンス内で作成される、仮想化されたオペレーティングシステム環境です。ゾーンは、アプリケーションを実行するための、分離され保護された環境を提供します。etherstub と VNIC は、Oracle Solaris の仮想化機能の一部にすぎません。VNIC または etherstub を Oracle Solaris で使用するように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。ゾーンの詳細は、『Oracle Solaris ゾーンの紹介 』を参照してください。

仮想ネットワークの仕組み

次の図は、システム内の仮想ネットワークとそのコンポーネントの仕組みを示しています。

図 1-2 仮想ネットワークの仕組み



この図は、1 枚の 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 つのゾーンに割り当てられます。`Zone 1`、`Zone 2`、および `Zone 3` は、システム内で使用するために構成された 3 つのゾーンです。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。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。両方ともシステムに接続できる場合、仮想スイッチの機能は物理スイッチの機能と同じです。

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。

また、これらのゾーンは、ゾーンに構成されているすべての VNIC が同じ VLAN の一部である場合は、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、`Zone 1` からその専用の `VNIC 1` を通して転送されます。そのトラフィックは次に、仮想スイッチを通して `VNIC 3` に転送されます。次に、`VNIC 3` がそのトラフィックを `Zone 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。

また、Oracle ではネットワーク仮想化の一部の側面 (たとえば、仮想データセンターの内部に仮想ネットワークを作成する機能) を管理するための Oracle Enterprise Manager Ops Center も提供しています。Oracle Enterprise Manager Ops Center の詳細は、ドキュメントライブラリ (http://www.oracle.com/pls/topic/lookup?ctx=oc122&id=OPCCM) を参照してください。

仮想拡張ローカルエリアネットワークの使用

仮想拡張ローカルエリアネットワーク (VXLAN) は、仮想ネットワークのスケーラビリティーとネットワーク分離を可能にするネットワーク仮想化テクノロジです。VXLAN は仮想ローカルエリアネットワーク (VLAN) の 4K の制限に対応し、スイッチなどの物理インフラストラクチャーでの仮想化の必要性も減らします。詳細は、第3章「仮想拡張ローカルエリアネットワークを使用することによる仮想ネットワークの構成」を参照してください。

エッジ仮想ブリッジングの使用

エッジ仮想ブリッジング (EVB) を使用すると、ホストがシステム上の仮想リンクに関連する情報を外部スイッチと交換できます。EVB は、ポートに関する情報を交換するために DCB を使用するのと同様に、ポートの背後にある仮想リンクに関する情報を交換するために使用されます。EVB の詳細は、第4章「エッジ仮想ブリッジングを使用したサーバーネットワークエッジの仮想化の管理」を参照してください。

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、仮想ネットワークの構築を検討してください。さまざまなアプリケーションを 2、3 のサーバーに統合することによって、使用可能なリソースをより適切に利用できます。そのあと、仮想ネットワークを使用して、アプリケーション間の接続を提供できます。

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、次のハードウェアリソースのためにサーバーを統合できます。

- 大量の帯域幅とハードウェアをサポートする強力な NIC (たとえば、NIC リングと仮想機能 (VF) のサポート)
- ランダムアクセスメモリー (RAM) と中央演算装置 (CPU) がより大きい、強力な物理マシン

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) 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

Oracle Solaris 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に提供す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。リソースが実際のニーズに応じて分散されるようにネットワークレーンを構成することによって、ネットワークパケットを処理する際のシステムの効率が向上します。ネットワークレーンの詳細は、197 ページの「データリンクとフローのネットワークトラフィックの統計情報のモニタリングの概要」を参照してください。

パケットを処理する際のシステムの効率を向上させるには、次のネットワークリソースを使用します。

- **帯域幅** – データリンクでサポートされるネットワークプロセスの実際のニーズに応じて、データリンクの帯域幅を制限できます。
- **優先順位** – パケットが処理される優先順位を設定できます。優先順位の高いパケットはほかのパケットの前に処理されるため、そのパケットの待機時間が短縮されます。
- **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。詳細は、171 ページの「NIC リングの管理」を参照してください。
- **CPU プール** – CPU のプールが作成され、特定のゾーンに関連付けられます。さらにこれらのプールをデータリンクに割り当てて、関連付けられたゾーンのネットワークプロセスを管理できます。詳細は、181 ページの「プールと CPU の管理」を参照してください。
- **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ことができます。詳細は、181 ページの「プールと CPU の管理」を参照してください。

システム上のネットワークリソースは、データリンクプロパティーまたはフローのいずれかを使用して管理できます。

データリンクプロパティーを使用することによるネットワークリソースの管理

データリンクプロパティーを使用してネットワークリソースを管理することで、パケットを処理する際のシステムの効率が向上します。データリンクプロパティーは、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、あとからデータリンクプロパティーを設定することもで

きます。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ー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決定できます。リソースを割り当てるための手順は、物理ネットワークだけでなく仮想ネットワーク環境にも当てはまります。データリンクプロパティーとそれらを構成する方法についての詳細は、169 ページの「データリンクプロパティ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。

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、単一の属性または属性の組み合わせに基づいてネットワークパケットを分類するためのカスタマイズされた方法です。フローを作成するための基盤として機能する属性は、ネットワークパケットのヘッダー内の情報から引き出されます。ネットワークリソースの管理のためのデータリンクプロパティーを設定したあと、フローを使用して、ネットワークパケットを処理するためのリソースの使用方法を詳細に制御できます。データリンクプロパティーを設定せずに、フローを単独で使用してネットワークリソースを管理できます。

リソースの管理にフローを使用するには、次の手順を実行します。

1. 単一の属性または属性の組み合わせに基づいた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ー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、フローに設定できるのは帯域幅と優先順位のプロパティーのみです。

フローの構成についての詳細は、186 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。

ネットワークリソースの管理の利点

ネットワークリソースの管理を利用することで、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。

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

♦ ♦ ♦ 第 2 章

仮想ネットワークの作成および管理

この章では、仮想ネットワークのコンポーネントの構成、仮想ネットワークの構築、および単一のシステム内の VNIC の管理のためのタスクについて説明します。仮想ネットワークの概要については、第1章「ネットワーク仮想化およびネットワークリソースの管理の概要」を参照してください。

この章の内容は、次のとおりです。

- 25 ページの「仮想ネットワークのコンポーネントの構成」
- 30 ページの「仮想ネットワークの構築」
- 40 ページの「VNIC の管理」
- 53 ページの「VNIC でのシングルルート I/O 仮想化の使用」

仮想ネットワークのコンポーネントの構成

Oracle Solaris では、VNIC と etherstub が仮想ネットワーク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、15 ページの「仮想ネットワークのコンポーネント」を参照してください。

仮想ネットワークの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
```

link VNIC が構成されているリンクの名前。

vid　VNIC を VLAN として作成する場合の VNIC の VLAN ID。VLAN ID を持つ VNIC を構成するには、28 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 11.2 でのネットワークデータリンクの管理 』の第 3 章「仮想ローカルエリアネットワークを使用した仮想ネットワークの構成」を参照してください。

VNIC　VNIC の名前。カスタマイズされた名前を作成する方法のガイドラインについては、『Oracle Solaris 11.2 でのネットワークコンポーネントの構成と管理 』の「有効なリンク名のための規則」を参照してください。

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ーを構成できます。これらのプロパティーの一覧については、`dladm`(1M) のマニュアルページを参照してください。特定のプロパティー変更は、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。データリンクと同様、VNIC には、必要に応じてさらに構成できるリンクプロパティーがあります。さまざまなタイプのリンクプロパティーの詳細は、22 ページの「データリンクプロパティ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。

`dladm create-vnic` コマンドを使用して作成できる VNIC に加え、システムでも、Oracle VM Server for SPARC の `vnet` 用の仮想ネットワーク I/O で役立つ、システムで作成された VNIC と呼ばれる VNIC を作成できます。システムで作成される VNIC は命名規則 `<entity>-<name>` に従います (`entity` は VNIC を作成したシステムエンティティーを指し、`name` はシステムエンティティー内での VNIC 名を指します)。ユーザーが作成する VNIC 名にハイフン (-) を含めることはできません。ハイフン (-) が含まれるのはシステムで作成された VNIC だけのため、システムで作成された VNIC とユーザーが作成した VNIC を区別するために役立ちます。システムで作成された VNIC の変更、名前変更、plumb、または削除を行うことはできません。詳細は、『*Oracle VM Server for SPARC 3.1 管理ガイド*』を参照してください。

`dlstat` および `snoop` コマンドを使用すると、システムで作成された VNIC 上のネットワークトラフィックをモニターできます。`flowadm` コマンドを使用して、システムで作成された VNIC 上にフローを作成することもできます。フローは、ネットワークリソースの管理だけでなく、ネットワークトラフィックの統計のモニターにも役立ちます。`flowstat` コマンドを使用すると、フロー上のネットワークトラフィックの統計をモニターできます。フローについての詳細は、188 ページの「フローの構成」を参照してください。

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

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

ここで、*etherstub* は作成する etherstub の名前です。

▼ 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、外部の LAN、およびインターネットと間の内部的なトラフィックをホストするには、各ゾーンに独自の VNIC 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 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」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、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てください。プライベート仮想ネットワークについては、14 ページの「仮想ネットワークの概要」を参照してください。プライベート仮想ネットワークを構成する方法についての詳細は、37 ページの「プライベート仮想ネットワークを構成する方法」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を作成する方法のガイドラインについては、『Oracle Solaris 11.2 でのネットワークコンポーネントの構成と管理 』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 link [-v vid] VNIC
   ```

   プライベート仮想ネットワーク用の VNIC を作成している場合は、*link* を *etherstub* に置き換えてください。VNIC を VLAN として作成している場合のみ、コマンドに -v オプションを含めます。VLAN としての VNIC の作成についての詳細は、28 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。

4. **VNIC 上に IP インタフェースを作成します。**

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

interface 前の手順で作成した VNIC。

5. **静的 IP アドレスを VNIC インタフェースに割り当てます。**

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

-a *address* IP アドレスを指定します (これはクラスレス相互ドメイン経路制御 (CIDR) 表記にすることができます)。

静的 IP アドレスは、IPv4 アドレスまたは IPv6 アドレスのどちらでもかまいません。詳細は、『Oracle Solaris 11.2 でのネットワークコンポーネントの構成と管理 』の「IPv4 インタフェースを構成する方法」を参照してください。

6. **(オプション) 作成された VNIC を確認します。**

```
# dladm show-link
```

例 2-1 VNIC の構成

この例は、データリンク net0 上に vnic1 を構成する方法を示しています。

```
# dladm create-vnic -l net0 vnic1
# ipadm create-ip vnic1
# ipadm create-addr -a 192.168.0.10/24 vnic1
# dladm show-link
LINK         CLASS     MTU    STATE    OVER
net0         phys      1500   up       --
vnic1        vnic      1500   up       net0
```

▼ VLAN ID を持つ VNIC を構成する方法

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。VNIC が VLAN の一部であり、その VLAN のトラフィックを受信する必要がある場合は、その VLAN の VLAN ID を VNIC に割り当てる必要があります。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、リンクプロパティー 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。通常の VLAN についての詳細は、『Oracle Solaris 11.2 でのネットワークデー

タリンクの管理 』の第 3 章「仮想ローカルエリアネットワークを使用した仮想ネットワークの構成」を参照してください。

この手順には、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ーを設定する手順のみが含まれています。vlan-announce プロパティー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、中間のポートやスイッチは、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 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」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. **(オプション) 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 プロパティーが 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。このプロパティーを 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。そのあと、これらのデバイスで VLAN トラフィックを受け入れ、転送できます。GVRP についての詳細は、"Configuring GVRP," in *Sun Ethernet Fabric Operating System*を参照してください。

4. **(オプション) gvrp-timeout プロパティーを設定して、VLAN のブロードキャスト間の待機時間を構成します。**

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

   time — gvrp-timeout プロパティーの値 (ミリ秒単位) を示します。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。このプロパティーを使用すると、この間隔を調整できます。

5. **(オプション) プロパティー vlan-announce および gvrp-timeout の値を表示します。**

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

例 2-2 VLAN としての VNIC の構成

この例は、VLAN ID が 123 のデータリンク net0 上に vnic0 という名前の VNIC を作成する方法と、VLAN 構成をネットワークに通知できるようにする方法を示しています。

```
# dladm create-vnic -l net0 -v 123 vnic0
# dladm set-linkprop -p vlan-announce=gvrp net0
# dladm set-linkprop -p gvrp-timeout=250 net0
# dladm show-linkprop -p vlan-announce,gvrp-timeout net0
LINK     PROPERTY        PERM   VALUE     EFFECTIVE    DEFAULT   POSSIBLE
net0     vlan-announce   rw     gvrp      gvrp         off       off,gvrp
net0     gvrp-timeout    rw     250       250          250       100-100000
```

出力には次の情報が表示されます。

LINK
: 物理データリンク (名前で識別されます)。

PROPERTY
: リンクのプロパティー。リンクには複数のプロパティーを設定できます。

PERM
: プロパティーのアクセス権 (次のいずれかを設定できます)。
  - ro は、リンクプロパティーの読み取り専用のアクセス権を示します。
  - rw は、リンクプロパティーの読み取りおよび書き込み権を示します。

VALUE
: 現在の (または永続的な) リンクプロパティーの値。値が設定されていない場合は、-- と表示されます。値が不明な場合は、? と表示されます。

DEFAULT
: リンクプロパティーのデフォルト値。リンクプロパティーにデフォルト値がない場合は、-- が表示されます。

POSSIBLE
: リンクプロパティーに設定できる値のコンマ区切りのリスト。指定可能な値が不明または無限の場合は、-- が表示されます。

仮想ネットワークの構築

仮想ネットワークを構築するには、ゾーンを作成する必要があります。システムのサポートに基づいて必要な数のゾーンを作成できます。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。

仮想ネットワークを構築するには、etherstub または VNIC を構成する必要があり、ゾーンも構成する必要があります。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築を完了するには、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います。

- 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、構成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、システムを越えて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必要があります。
- システムの物理インタフェースには、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。

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。

仮想ネットワークの構成のデモについては、Oracle Solaris での仮想ネットワークの構成 - 第 1 部 (http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/VirtualDemo_Part1/VirtualDemo_Part1.htm) およびOracle Solaris での仮想ネットワークの構成 - 第 2 部 (http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/VirtualDemo_Part2/VirtualDemo_Part2.htm) のページを参照してください。

▼ 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。ゾーンを構成する方法についての詳細は、『Oracle Solaris ゾーンの作成と使用 』の第 1 章「非大域ゾーンの計画および構成方法」を参照してください。

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **VNIC を構成します。**

   詳細は、27 ページの「VNIC と etherstub を構成する方法」を参照してください。

3. **ゾーンを作成します。**

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

zone ゾーンの名前に示します。

必ず、以前にゾーンの物理インタフェースとして作成した VNIC を割り当ててください。デフォルトでは、ゾーンの `ip-type` パラメータは `exclusive` に設定されます。

4. **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:zone> verify
zonecfg:zone> commit
zonecfg:zone> exit
```

5. **ゾーンをインストールします。**

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

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。**

```
global# 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できます。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。

または、`anet` リソースという自動 VNIC を持つ排他的 IP ゾーンを構成することもできます。詳細は、『Oracle Solaris ゾーンの作成と使用 』の「ゾーンの構成方法」を参照してください。

例 2-3 仮想ネットワークのためのゾーンの構成

この例では、zone1 が仮想ネットワーク用に作成され、vnic1 が物理インタフェースとして接続されます。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ることに注意してください。

```
global # zonecfg -z zone1
zonecfg:zone1> create
zonecfg:zone1> set zonepath=/export/home/zone1
zonecfg:zone1> set autoboot=true
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
.
.
.
global# zoneadm -z zone1 boot

global# zlogin -C zone1
```

ネットワークを構成するには、次の情報を指定します。

```
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。こ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 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **VNIC を作成します。**

   ```
   global# dladm create-vnic -l link VNIC
   ```

   この手順のあとで、VNIC のインタフェースを構成します。

3. **ゾーンの IP タイプを shared から exclusive に変更します。**

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

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:zone> remove net physical=NIC
   zonecfg:zone> add net
   zonecfg:zone:net> set physical=VNIC
   zonecfg:zone:net> end
   ```

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:zone> verify
   zonecfg:zone> commit
   zonecfg:zone> exit
   ```

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. **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。**

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

9. **静的 IP アドレスまたは動的ホスト構成プロトコル (DHCP) IP アドレスを持つ VNIC を構成します。**

   - **静的 IP アドレスを割り当てます。**

     ```
     zone# 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。

   - **DHCP IP アドレスを割り当てます。**

```
zone# ipadm create-addr -T dhcp interface
```

10. **ゾーンを終了します。**

```
zone# exit
```

11. **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 2-4 VNIC を使用するためのゾーンの再構成

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 exclusive 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:zone2> set ip-type=exclusive
zonecfg:zone2> remove net physical=net0
zonecfg:zone2> add net
zonecfg:zone2:net> set physical=vnic2
zonecfg:zone2:net> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> 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# pfedit /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
```

▼ ゾーンで一時的に VNIC を作成する方法

VNIC は、リンクを *zone*/*link* として指定することで、大域ゾーンから非大域ゾーン内に直接作成できます。この方法では、非大域ゾーンの名前空間に VNIC が直接作成されます。VNIC が一時的であることを指定するには、`-t` オプションを使用します。一時的な VNIC はゾーンの次回のリブートまで保持されます。大域ゾーンやほかの非大域ゾーンは、同じ名前の VNIC を持つこともできます。この方法を使用すると、VNIC をほんの一時的に作成できます。

一時的な VNIC の作成に加えて、VLAN と IP over InfiniBand (IPoIB) のパーティションを一時的に作成することもできます。詳細な手順については、`dladm`(1M) のマニュアルページを参照してください。

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **大域ゾーンから非大域ゾーンを作成してブートします。**

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

3. **非大域ゾーン用の一時的な VNIC を作成します。**

   ```
   global# dladm create-vnic -t -l link zone/VNIC
   ```

`-t`	VNIC を一時的なものとして指定します。一時的な VNIC は *zone* の次回のリブートまで保持されます。VNIC を非大域ゾーンの名前空間内に作成する場合は、このオプションを指定する必要があります。
`-l`	リンクを指定します (物理リンクまたは etherstub を指定できます)。

   大域ゾーンから非大域ゾーン内に VLAN または IPoIB パーティションを作成するために使用するコマンド構文の例については、例2-5「ゾーンでの VNIC、VLAN、および IP-over-IB パーティションの一時的な作成」 を参照してください。

4. **ゾーンに VNIC が作成されたことを確認します。**

   ```
   global# dladm show-link -Z
   ```

5. **ゾーンにログインします。**

   ```
   global# zlogin zone
   ```

6. **VNIC が正常に作成されたことを確認します。**

```
zone# dladm show-link
```

例 2-5 ゾーンでの VNIC、VLAN、および IP-over-IB パーティションの一時的な作成

次の例は、大域ゾーンから非大域ゾーン内に `vnic1` という名前の VNIC を作成する方法を示しています。

```
global# zoneadm -z zone1 boot
global# dladm create-vnic -t -l net0 zone1/vnic1
global# dladm show-link -Z
LINK                ZONE      CLASS     MTU    STATE    OVER
net0                global    phys      1500   up        --
zone1/vnic1         zone1     vnic      1500   down     net0
```

次の例は、`zone1` からの `dladm show-link` コマンドの出力を示しています。

```
zone1# dladm show-link
LINK                CLASS     MTU    STATE    OVER
vnic1               vnic      1500   down     ?
```

次の例は、大域ゾーンから非大域ゾーン内に `vnlan3` という名前の VLAN を作成する方法を示しています。

```
global# dladm create-vlan -t -l net0 -v 3 zone1/vlan3
```

-v オプションは、Ethernet リンク上の VLAN の VLAN-ID を指定します。

次の例は、大域ゾーンから非大域ゾーン内に `part1` という名前の IPoIBV パーティションを作成する方法を示しています。

```
global# dladm create-part -t -l net1 -P FFFF zone1/part1
```

-P オプションは、パーティションリンクを作成するために使用するパーティションキーを指定します。

▼ プライベート仮想ネットワークを構成する方法

この手順では、プライベート仮想ネットワークを作成して、システムを越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方法について説明します。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越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 11.2 でのネットワークのセキュリティー保護 』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、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、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 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」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
   ```

   この手順のあとで、VNIC のインタフェースを構成します。

4. **ゾーンに VNIC を割り当てます。**

   ```
   global# zonecfg -z zone
   zonecfg:zone> add net
   zonecfg:zone:net> set physical=VNIC
   zonecfg:zone:net> end
   ```

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:zone> verify
   zonecfg:zone> commit
   zonecfg:zone> exit
   ```

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. **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。**

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

9. 静的 IP アドレスまたは DHCP IP アドレスを持つ VNIC を構成します。

   ■ 静的 IP アドレスを割り当てます。

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

   ■ DHCP IP アドレスを割り当てます。

   ```
   zone# ipadm create-addr -T dhcp interface
   ```

10. ゾーンを終了します。

    ```
    zone# exit
    ```

11. 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。

12. 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。

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

    注記 - Oracle Solaris では、プライマリインタフェースは NIC の物理的なデータリンクです。

13. 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。

14. IP フィルタサービスを起動して NAT を有効にします。

    ```
    global# svcadm enable network/ipfilter
    ```

15. ゾーンをリブートします。

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

例 2-6 プライベート仮想ネットワークの構成

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、プライベート仮想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、ゾーンに 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# zoneadm -z zone3 reboot
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# pfedit /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# pfedit /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 reboot
```

VNIC の管理

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。VNIC の基本構成を実行する方法について詳しくは、27 ページの「VNIC と etherstub を構成する方法」を参照してください。

VLAN ID、MAC アドレス、および VNIC のベースとなるデータリンクを変更できます。ベースとなるデータ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。データリンク上のすべての VNIC の属性を全体的変更するか、指定した VNIC のみの属性を選択的に変更できます。

このセクションの内容は次のとおりです。

- 41 ページの「VNIC の表示」

- 45 ページの「VNIC の VLAN ID の変更」
- 46 ページの「VNIC の MAC アドレスの変更」
- 48 ページの「VNIC の移行」
- 50 ページの「VNIC の削除」

VNIC 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。

例 2-7 システム上の VNIC の表示

```
# dladm show-vnic
LINK     OVER    SPEED       MACADDRESS        MACADDRTYPE        VIDS
vnic1    net0    1000        2:8:20:c2:39:38   random             123
vnic2    net0    1000        2:8:20:5f:84:ff   random             456
```

出力には次の情報が表示されます。

`LINK`
仮想データリンク (名前で識別されます)。

`OVER`
VNIC が構成されている物理または仮想データリンク。

`SPEED`
VNIC の最大速度 (M ビット/秒単位)。

`MACADDRESS`
VNIC の MAC アドレス。

`MACADDRTYPE`
VNIC の MAC アドレスのタイプ (次のいずれかを設定できます)。

- `random` – VNIC に割り当てられたランダムなアドレス
- `factory` – VNIC によって使用される NIC のファクトリ MAC アドレス
- `fixed` – ユーザーが割り当てる MAC アドレス

`VID`
VNIC の VLAN ID。

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ができます。たとえば、`dladm show-link` コマンドでは、VNIC とその他のデータリンクが表示されます。`dladm show-linkprop` コマンドを使用すると、VNIC のプロパティーを表示できます。

単一の VNIC のデータリンクプロパティーに関する情報を取得するには、次のコマンド構文に VNIC を指定します。

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

例 2-8 ゾーンに接続されている VNIC の表示

この例では、ゾーンに接続されたプライマリデータリンクと VNIC の情報が表示されます。プライマリデータリンク net0 は大域ゾーンに接続されています。VNIC、vnic1 と vnic2 は、それぞれ zone1 と zone2 に接続されています。

```
# dladm show-link -Z
LINK                ZONE      CLASS     MTU    STATE    OVER
net0                global    phys      1500   up        --
zone1/vnic1         zone1     vnic      1500   up       net0
zone2/vnic2         zone2     vnic      1500   up       net0
```

複数の MAC アドレスを持つ VNIC を表示

Oracle VM Server for SPARC のシステムで作成された VNIC と、Oracle Solaris カーネルゾーンの anet リソースには、複数の MAC アドレスが関連付けられます。Oracle VM Server for SPARC でゲストドメイン内部の VNIC とゾーンをサポートするには、alt-mac-addrs プロパティーを使用して vnet を作成する必要があります。この場合、システムは自動的に複数の MAC アドレスを持つ VNIC を作成します。これらの複数の MAC アドレスは、作成した vnet から取得されます。詳細は、『*Oracle VM Server for SPARC 3.1 管理ガイド*』を参照してください。

カーネルゾーン内部のゾーンまたは VNIC をサポートするには、複数の MAC アドレスを持つ anet リソースを構成します。カーネルゾーンでのネットワークアクセスのために作成された anet リソースに複数の MAC アドレスを指定するには、zonecfg コマンドを使用します。詳細は、solaris-kz(5) のマニュアルページを参照してください。カーネルゾーンの構成についての詳細は、『Oracle Solaris カーネルゾーンの作成と使用 』を参照してください。

VNIC に複数の MAC アドレスが関連付けられている場合は、仮想ネットワークドライバによって 1 つの MAC アドレスが使用されます。残りの MAC アドレスを使用して、カーネルゾーンまたはゲストドメイン内に VNIC を作成できます。たとえば、VNIC が 3 つの MAC アドレスに関連付けられている場合は、1 つの MAC アドレスが仮想ネットワークドライバに割り当てられています。そのため、残りの 2 つの MAC アドレスを使用して作成できる VNIC は 2 つだけです。

次のコマンドを使用すると、VNIC に関連付けられた複数の MAC アドレスを表示できます。

```
# dladm show-vnic -m
```

例 2-9 カーネルゾーン内の複数の MAC アドレスを持つ VNIC の表示

```
# dladm show-vnic -m
LINK              OVER      MACADDRESSES      MACADDRTYPES     VIDS
gz_vnic0          net0      2:8:20:d7:27:9d   random            0
zone1/net0        net0      2:8:20:70:52:9    random            0
                            2:8:20:c9:d:4c    fixed
                            2:8:20:70:db:3    random
zone1/net1        net0      0:1:2:3:4:5       fixed             0
                            0:1:2:3:4:6       fixed
```

この例では、カーネルゾーン zone1 に net0 と net1 の 2 つの anet リソースが存在します。どちらのリソースにも複数の MAC アドレスが構成されています。そのため、カーネルゾーン zone1 内では、データリンク net0 に関連付けられた仮想 NIC ドライバ zvnet の最上位に最大 2 つの VNIC を作成できます。データリンク net1 に関連付けられた仮想 NIC ドライバ zvnet の最上位に作成できる VNIC は 1 つだけです。

例 2-10 複数の MAC アドレスを持つ、システムで生成された VNIC の表示

```
# dladm show-vnic -m
LINK              OVER      MACADDRESSES      MACADDRTYPES     VIDS
ldoms-vsw0.vport0 net1      0:14:4f:fb:e1:8f  fixed            0,21
                            0:14:4f:f8:6b:9   fixed
                            0:14:4f:fa:48:7f  fixed
ldoms-vsw0.vport1 net1      0:14:4f:f9:1b:8d  fixed            45,44
                            0:14:4f:f9:27:4   fixed
```

この例では、ldoms-vsw0.vport0 に関連付けられたゲストドメインの仮想ネットワークドライバ vnet の最上位に、最大 2 つの VNIC を作成できます。ldoms-vsw0.vport1 に関連付けられた仮想 NIC ドライバ vnet の最上位には、最大 1 つの VNIC を作成できます。

データリンクの物理および仮想リンク状態の表示

データリンクの物理リンク状態は、物理デバイスに外部ネットワークとの接続があるかどうかを識別します。ケーブルが接続され、そのケーブルの反対側にあるポートの状態が up の場合は、物理デバイスに外部ネットワークとの接続があります。

次のコマンドを使用すると、データリンクの物理リンク状態を表示できます。

```
# dladm show-phys [link]
```

```
# dladm show-ether [link]
```

詳細については、dladm(1M) のマニュアルページを参照してください。

例 2-11 データリンクの物理リンク状態の表示

次の例では、dladm show-phys コマンドを使用して、システム上のデータリンクの物理リンク状態を表示します。

```
# dladm show-phys
LINK       MEDIA        STATE      SPEED  DUPLEX    DEVICE
net1       Ethernet     down       0      unknown   e1000g1
net2       Ethernet     down       0      unknown   e1000g2
net3       Ethernet     down       0      unknown   e1000g3
net0       Ethernet     up         1000   full      e1000g0
```

次の例では、dladm show-enter コマンドを使用して、システム上のデータリンクの物理リンク状態を表示します。

```
# dladm show-ether
LINK       PTYPE    STATE    AUTO  SPEED-DUPLEX     PAUSE
net1       current  down     yes   0M               bi
net2       current  down     yes   0M               bi
net3       current  down     yes   0M               bi
net0       current  up       yes   1G-f             bi
```

NIC 上に複数の VNIC が作成されると、同じ VLAN 上にあるときに VNIC とプライマリデータリンクが通信できるように、内部的に仮想スイッチが作成されます。物理データリンクが外部ネットワークに接続されていない場合でも、これらのデータリンクは互いに通信できます。これにより、データリンクの仮想リンク状態が形成され、その状態は up、down、unknown のいずれかになります。データリンクの仮想リンク状態は、物理ケーブルが抜かれている場合でも、データリンクがシステム内の内部ネットワークに接続されているかどうかを識別します。

データリンクの仮想リンク状態を表示するには、次のコマンドを使用します。

```
# dladm show-link [link]
```

例 2-12 データリンクの仮想リンク状態の表示

この例では、システム上のデータリンクの仮想リンク状態が表示されます。

```
# dladm show-link
LINK        CLASS     MTU    STATE    OVER
net0        phys      1500   up       --
net2        phys      1500   down     --
net4        phys      1500   down     --
net1        phys      1500   up       --
net5        phys      1500   up       --
vnic0       vnic      1500   up       net5
vnic1       vnic      1500   up       net5
vnic2       vnic      1500   up       net1
```

VNIC の VLAN ID の変更

VNIC を VLAN として構成できます。VNIC に特定の VLAN のトラフィックをホストさせるときは、データリンク上の VNIC の VLAN ID を変更する必要があります。

使用する `dladm` サブコマンドは、VLAN または VLAN として構成された VNIC のどちらを変更するかによって異なります。

- `dladm create-vlan` コマンドを使用して作成された VLAN の場合は、`dladm modify-vlan` コマンドを使用します。これらの VLAN を表示するには、`dladm show-vlan` コマンドを使用します。
- `dladm create-vnic` コマンドを使用して作成された VLAN の場合は、`dladm modify-vnic` コマンドを使用します。これらの VNIC (VLAN ID を持つものを含む) を表示するには、`dladm show-vnic` コマンドを使用します。

データリンク上に構成されている単一の VNIC または複数の VNIC の VLAN ID を変更できます。同じ VLAN ID を持つすべての VNIC を構成することで、VNIC の VLAN ID をグループとして変更することもできます。

- データリンク上に 1 つの VNIC だけが構成されている場合は、次のコマンド構文を使用して VNIC の VLAN ID を変更します。

  ```
  # dladm modify-vnic -v vid -L link
  ```

  ここで、*vid* は VNIC に割り当てる新しい VLAN ID です。

例 2-13 データリンク上の VNIC の VLAN ID の変更

この例では、データリンク `net0` 上に構成されている `vnic0` の VLAN ID が変更されます。

```
# dladm modify-vnic -v 123 -L net0
# dladm show-vnic
LINK      OVER     SPEED      MACADDRESS        MACADDRTYPE      VIDS
vnic0     net0     1000       2:8:20:c2:39:38   random           123
```

- データリンク上に複数の VNIC が構成されている場合、次のコマンド構文を使用して VNIC の VLAN ID を変更します。

  ```
  # dladm modify-vnic -v vid VNIC
  ```

  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。

例 2-14 データリンク上の複数の VNIC の VLAN ID の変更

この例では、vnic0、vnic1、および vnic2 の VLAN ID が変更されます。

```
# dladm modify-vnic -v 123 vnic0
# dladm modify-vnic -v 456 vnic1
# dladm modify-vnic -v 789 vnic2
# dladm show-vnic
LINK      OVER     SPEED        MACADDRESS          MACADDRTYPE         VIDS
vnic0     net0     1000         2:8:20:c2:39:38     random              123
vnic1     net0     1000         2:8:20:5f:84:ff     random              456
vnic2     net0     1000         2:8:20:5f:84:ff     random              789
```

- それぞれの VNIC が異なるデータリンク上に構成されている場合は、次のコマンド構文を使用して、VNIC の VLAN ID をグループとして変更します。

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

例 2-15 VNIC の VLAN ID のグループとしての変更

この例では、vnic0、vnic1、および vnic2 の VLAN ID がグループとして変更されます。これらの VNIC は、それぞれ、データリンク net0、net1、および net2 上に構成されています。

```
# dladm modify-vnic -v 123 vnic0,vnic1,vnic2
# dladm show-vnic
LINK      OVER     SPEED        MACADDRESS          MACADDRTYPE         VIDS
vnic0     net0     1000         2:8:20:c2:39:38     random              123
vnic1     net1     1000         2:8:20:5f:84:ff     random              123
vnic2     net2     1000         2:8:20:5f:84:ff     random              123
```

VNIC の MAC アドレスの変更

ユーザーが作成する VNIC は、MAC アドレスを 1 つしか持つことができません。この MAC アドレスは、dladm modify-vnic コマンドを使用して変更できます。カーネルゾーン用に作成された VNIC は、1 つ以上の MAC アドレスを持つように構成できます。

データリンク上で構成された VNIC の既存の MAC アドレスを変更できます。すべての VNIC の MAC アドレスを変更することも、指定した VNIC の MAC アドレスを選択的に変更することもできます。VNIC の VLAN ID と MAC アドレスを同時に変更することもできます。

- VNIC の MAC アドレスを変更するには、次のコマンド構文を使用します。

  ```
  # dladm modify-vnic -m MAC-address VNIC
  ```

ここで、*MAC-address* は VNIC に割り当てる新しい MAC アドレスです。

例 2-16 VNIC の MAC アドレスの変更

この例では、`vnic0` に特定の MAC アドレスが割り当てられます。

```
# dladm modify-vnic -m 3:8:20:5f:84:ff vnic0
# dladm show-vnic
LINK      OVER     SPEED         MACADDRESS         MACADDRTYPE         VIDS
vnic0     net0     1000         3:8:20:5f:84:ff     fixed               0
```

- データリンク上のすべての VNIC の MAC アドレスを変更するには、次のコマンド構文を使用します。

  `# dladm modify-vnic -m random -L` *link*

  このコマンド構文では、`-m random` オプションは `-m auto` オプションと同等です。MAC アドレスは自動的に、VNIC にランダムに割り当てられます。

例 2-17 データリンク上のすべての VNIC の MAC アドレスの変更

この例では、データリンク `net0` 上に構成されたすべての VNIC の MAC アドレスが、自動的にランダムに変更されます。

```
# dladm modify-vnic -m random -L net0
# dladm show-vnic
LINK      OVER     SPEED        MACADDRESS          MACADDRTYPE       VIDS
vnic0     net0     1000         2:8:20:22:9d:bb     random             0
vnic1     net0     1000         2:8:20:72:2e:9      random             0
vnic2     net0     1000         2:8:20:2f:e5:83     random             0
```

- VNIC の MAC アドレスを選択的に変更するには、次のコマンド構文を使用します。

  `# dladm modify-vnic -m random` *VNIC,VNIC*`,[...]`

  全体的な変更と選択的な変更のどちらの場合も、`-m` オプションに `random` を指定します。

例 2-18 VNIC の MAC アドレスの選択的な変更

この例では、データリンク `net0` 上に構成されている `vnic0` および `vnic2` の MAC アドレスが選択的に変更されます。

```
# dladm modify-vnic -m random vnic0,vnic2
# dladm show-vnic
LINK      OVER     SPEED         MACADDRESS         MACADDRTYPE        VIDS
```

```
vnic0     net0     1000        2:8:20:2f:e5:83     random             0
vnic1     net0     1000        2:8:20:5f:84:ff     fixed              0
vnic2     net0     1000        2:8:20:2f:e5:83     random             0
```

- VNIC の VLAN ID と MAC アドレスを同時に変更するには、次のコマンド構文を使用します。

  # dladm modify-vnic -m random -v *vid VNIC*

注意 - VNIC の複数の属性を全体的に変更すると、VNIC で予期しない動作が発生する可能性があります。代わりに、VNIC の複数の属性を個別に変更してください。

例 2-19 VNIC の VLAN ID と MAC アドレスの変更

この例では、vnic0 の VLAN ID と MAC アドレスが同時に変更されます。

```
# dladm modify-vnic -m random -v 123 vnic0
# dladm show-vnic vnic0
LINK      OVER     SPEED  MACADDRESS       MACADDRTYPE    VIDS
vnic0     net0     1000   2:8:20:2f:e5:83  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。ベースとなるデータリンクは、物理リンク、リンクアグリゲーション、または etherstub です。

通常は、次のいずれかの状況で VNIC を移行します。

- 既存の NIC を新しい NIC と交換する必要がある場合
- ターゲットの NIC の帯域幅が既存の NIC より広い場合
- ターゲットの NIC がハードウェアにラージレシーブオフロード (LRO)、ラージセグメントオフロード (LSO)、チェックサムなどの特定の機能を実装する場合

VNIC を正常に移行するには、VNIC の移動先のターゲットのデータリンクが、その VNIC のデータリンクプロパティーに対応している必要があります。これらのプロパティーがサポートされていない場合、移行は失敗し、ユーザーに通知されます。移行が成功したあと、ターゲットデータリンク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ーは変更される可能性があります。これらのプロパティーの値は、VNIC の移行先のデータリンクから継承されます。データリンク上に構成されているすべての VNIC を移行することも、指定された VNIC を選択的に移行することもできます。VNIC を移行し、その VLAN ID を同時に変更することもできます。

- ソースリンク上に構成されているすべての VNIC をターゲットリンクに移行するには、次のコマンド構文を使用します。

  ```
  # dladm modify-vnic -l target-link -L source-link
  ```

  -l *target-link* VNIC が移行されるリンクを示します

  -L *source-link* VNIC が以前に構成されたリンクを示します

例 2-20 ソースリンクからターゲットリンクへのすべての VNIC の移行

この例では、ソースリンク ether0 からのすべての VNIC がターゲットリンク net1 に移動されます。

```
# dladm modify-vnic -l net1 -L ether0
# dladm show-vnic
LINK      OVER      SPEED        MACADDRESS          MACADDRTYPE        VIDS
vnic0     net1      1000        2:8:20:c2:39:38      random             321
vnic1     net1      1000        2:8:20:5f:84:ff      random             656
vnic2     net1      1000        2:8:20:5f:84:ff      random              0
```

- ソースリンク上に構成されている指定された VNIC をターゲットリンクに移行するには、次のコマンド構文を使用します。

  ```
  # dladm modify-vnic -l target-link VNIC,VNIC,[...]
  ```

  VNIC の選択的な移行を実行するには、ターゲットリンクだけを指定する必要があります。

例 2-21 ソースリンクからターゲットリンクへの指定した VNIC の移行

この例では、vnic0、vnic1、および vnic2 が、ソースリンク net0 からターゲットリンク net1 に選択的に移動されます。

```
# dladm modify-vnic -l net1 vnic0,vnic1,vnic2
# dladm show-vnic
LINK      OVER      SPEED        MACADDRESS          MACADDRTYPE        VIDS
vnic0     net1      1000        2:8:20:c2:39:38      random             321
vnic1     net1      1000        2:8:20:5f:84:ff      random             656
```

```
vnic2      net1      1000        2:8:20:5f:84:ff      random              0
vnic3      net0      1000        2:8:20:5f:84:ff      random            345
```

- ソースリンク上に構成されている VNIC の VLAN ID を変更して、同時にターゲットリンクに移行するには、次のコマンド構文を使用します。

  `# dladm modify-vnic -l` *target-link* `-v` *vid VNIC*

  新しい VLAN ID を割り当てするには、一度に 1 つの VNIC を移行する必要があります。

例 2-22 VNIC の VLAN ID の移行と変更

この例では、vnic0、vnic1、および vnic2 がターゲットデータリンク net1 に移行されます。移行では、すべての VNIC の VLAN ID も同時に変更されます。

```
# dladm modify-vnic -l net1 -v 123 vnic0
# dladm modify-vnic -l net1 -v 456 vnic1
# dladm modify-vnic -l net1 -v 789 vnic2
# dladm show-vnic
LINK       OVER      SPEED        MACADDRESS          MACADDRTYPE        VIDS
vnic0      net1      1000        2:8:20:c2:39:38      random            123
vnic1      net1      1000        2:8:20:5f:84:ff      random            456
vnic2      net1      1000        2:8:20:5f:84:ff      random            789
```

VNIC をソースリンクからターゲットリンクに移行する際、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。例2-22「VNIC の VLAN ID の移行と変更」を参照してください。

ただし、VNIC がソースリンクからのファクトリ MAC アドレスを使用している場合は、MAC アドレスが変更されます。移行中に MAC アドレスを指定しないと、VNIC のファクトリ MAC アドレスがランダムに割り当てられた MAC アドレスに置き換えられます。移行中に -m を付けて MAC アドレスを指定すると、VNIC のファクトリ MAC アドレスが指定された MAC アドレスで置き換えられます。

複数の MAC アドレスが、カーネルゾーンで作成された VNIC に関連付けられています。カーネルゾーンで作成された VNIC を移行すると、VNIC に関連付けられた複数の MAC アドレスがすべて、ターゲットの NIC に移行されます。

VNIC の削除

このセクションでは、VNIC を削除する方法について説明します。

▼ VNIC を削除する方法

1. **管理者になります。**

2. **(オプション) VNIC がビジー状態かどうかをチェックします。**

   VNIC を削除できるのは、ビジー状態ではない場合のみです。VNIC はさまざまな理由でビジー状態の場合があります。VNIC がビジー状態かどうかを確認するには、次の手順を実行する必要があります。

   - **VNIC が plumb され、IP アドレスに関連付けられているかどうかを確認します。**

     ```
     # ipadm show-if
     # ipadm show-addr
     ```

     VNIC が plumb され、IP アドレスと関連付けられている場合は、IP インタフェースを削除します。

     ```
     # ipadm delete-ip interface
     ```

   - **VNIC 上に構成されているフローがあるかどうかを確認します。**

     ```
     # flowadm
     ```

     フローが VNIC 上に構成されている場合は、フローを削除します。

     ```
     # flowadm remove-flow flowname
     ```

   - **VNIC がゾーンに割り当てられているかどうかを確認します。**

     ```
     # dladm show-link -Z
     ```

     ゾーンに接続されている VNIC を削除する方法についての詳細は、52 ページの「ゾーンに接続された VNIC を削除する方法」を参照してください。

   - **VNIC がシステムによって作成されているかどうかを確認します。**

     ```
     # dladm show-vnic
     ```

     ハイフン (-) が含まれるのはシステムで作成された VNIC だけのため、システムで作成された VNIC とユーザーが作成した VNIC を区別するために役立ちます。システムで作成された VNIC の変更、名前変更、plumb、または削除を行うことはできません。

   - **VNIC がスヌープされているかどうかを確認します。**

     ```
     # snoop
     ```

```
# tshark
```

VNIC が `snoop` コマンドを使用してスヌープされている場合は、プロセスを強制終了します。

```
# pkill snoop
```

VNIC が `tshark` コマンドを使用してスヌープされている場合は、プロセスを強制終了します。

```
# pkill tshark
```

3. **VNIC を削除します。**

```
# dladm delete-vnic VNIC
```

▼ ゾーンに接続された VNIC を削除する方法

この手順では、VNIC がゾーンに接続されていると仮定します。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. **ゾーンを停止します。**

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

注記 - 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# zoneadm -z zone boot
```

例 2-23 ゾーンに接続された 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# zoneadm -z zoneB boot
```

VNIC でのシングルルート I/O 仮想化の使用

Oracle Solaris 11.2 リリース以降、`dladm` コマンドを使用して、シングルルート I/O 仮想化 (SR-IOV) をサポートするネットワークデバイスを管理できます。SR-IOV は、仮想マシン間での Peripheral Component Interconnect Express (PCIe) デバイスの効率的な共有を可能にする標準です。これはハードウェアに実装されます。詳細は、『Oracle Solaris 11.2 デバイスドライバの記述 』の「SR-IOV の概要」を参照してください。

データリンクの SR-IOV モードの有効化

Oracle Solaris では、SR-IOV をサポートするネットワークデバイスの仮想機能 (VF) を VNIC または VLAN に関連付けることができます。VF VNIC は専用の VF を所有する VNIC です。VF VNIC は、リソースの共有の点で通常の VNIC とは異なります。通常の VNIC はほかの通常の VNIC とリソースを共有する必要がありますが、VF VNIC はリソースを共有する必要がありません。各 VF は VF VNIC 用の個別のハードウェアリソースです。

VF VNIC を作成できるのは、SR-IOV モードをサポートするデータリンク上だけです。デフォルトでは、データリンクの SR-IOV モードは無効になっています。`iov` プロパティーを `on` に設定することで、データリンクの SR-IOV モードを有効にすることができます。データリンクの SR-IOV モードを有効にしたあとの VF VNIC の作成の詳細は、54 ページの「VF VNIC の作成」を参照してください。

dladm show-linkprop コマンドでリンクプロパティー iov を指定することで、データリンクの SR-IOV モードを確認できます。出力の EFFECTIVE 列の下の値が off の場合、データリンクの SR-IOV モードは無効になっています。

次の例は、データリンク net0 の SR-IOV モードをどのようにして確認できるかを示しています。

```
# dladm show-linkprop -p iov net0
LINK     PROPERTY    PERM     VALUE     EFFECTIVE     DEFAULT      POSSIBLE
net0     iov         rw       auto      off           auto         auto,on,off
```

この例では、データリンク net0 の SR-IOV モードは無効になっています。出力には次の情報が表示されます。

VALUE
: iov リンクプロパティーに対して設定した値を指定します。iov リンクプロパティーを変更しなかった場合、iov リンクプロパティーのデフォルト値は auto です。値 auto は、特定の物理データリンクで SR-IOV モードがデフォルトで有効になっているかどうかを OS が確認することを意味します。

EFFECTIVE
: データリンクの実際の SR-IOV モード。デフォルトでは、SRIOV 対応のすべての NIC で、EFFECTIVE 列の下に値 off が表示されます。

次のように iov プロパティーを on に設定することで、データリンク net0 の SR-IOV モードを有効にすることができます。

```
# dladm set-linkprop -p iov=on net0
# dladm show-linkprop -p iov net0
LINK     PROPERTY    PERM     VALUE     EFFECTIVE     DEFAULT      POSSIBLE
net0     iov         rw       on        on            auto         auto,on,off
```

同様に、iov リンクプロパティーを off に設定することで、データリンクの SR-IOV モードを無効にすることができます。dladm コマンドの詳細は、dladm(1M) のマニュアルページを参照してください。

VF VNIC の作成

データリンク上に VF VNIC を作成するには、データリンクの SR-IOV モードを有効にする必要があります。詳細は、53 ページの「データリンクの SR-IOV モードの有効化」を参照してください。データリンクの SR-IOV モードを有効にしたあとで、dladm create-vnic コマンドを使用して VNIC を作成すると、VF が自動的に VNIC に割り当てられます。同様に、dladm create-vlan コマンドを使用して VLAN を作成すると、VF が自動的に VLAN に割り当てられます。

dladm create-vnic または dladm create-vlan コマンドで iov VNIC リンクプロパティーを指定することによって、VF を VNIC または VLAN のどちらに割り当てる必要があるかを明示的に指定することもできます。

VF VNIC を明示的に作成するには、次のコマンド構文を使用します。

```
# dladm create-vnic [-p iov=value] -l link VNIC
```

VF VNIC を作成する場合、iov VNIC リンクプロパティーの指定はオプションです。iov VNIC リンクプロパティーを指定しない場合は、デフォルト値 inherit がこのプロパティーに割り当てられます。iov VNIC リンクプロパティーに次の値を指定できます。

inherit
: iov VNIC リンクプロパティーのデフォルト値。ベースとなるデータリンクの有効な iov プロパティー値に基づいて VF を割り当てる必要があるかどうかを決定します。
  - off – VNIC 用の VF を割り当てません。
  - on – VNIC 用の VF の割り当てが試行されます。可能でない場合は、通常の VNIC が作成されます。

on
: VF を割り当てます。VF が見つからない場合、VNIC の作成は失敗します。

off
: VF のない VNIC を作成します。

データリンクプロパティーの有効値は、データリンクに対して dladm show-linkprop コマンドを使用するときに EFFECTIVE 列の下に表示される値です。

iov VNIC リンクプロパティーとほかのデータリンクプロパティーの違いは、iov VNIC リンクプロパティーは VNIC または VLAN を作成するときだけ指定できるという点です。VNIC または VLAN を作成したあとで iov VNIC リンクプロパティーを変更することはできません。

iov VNIC リンクプロパティーには、VF が VNIC または VLAN のどちらに割り当てられているかを示す有効値があります。EFFECTIVE 列の下の値 on は VF が割り当てられていることを意味し、EFFECTIVE 列の下の値 off は VF が割り当てられていないことを意味します。

例 2-24 VF VNIC の作成

次の例は、iov VNIC リンクプロパティーを明示的に指定することで、データリンク net0 上の VF VNIC vfvnic1 と通常の VNIC vnic1 を作成する方法を示しています。この例では、データリンク net0 の SR-IOV モードが有効になっていると仮定しています。

```
# dladm show-linkprop -p iov net0
```

```
LINK     PROPERTY  PERM    VALUE  EFFECTIVE  DEFAULT    POSSIBLE
net0     iov       rw      on     on         auto       auto,on,off
# dladm create-vnic -l net0 vfvnic1
# dladm show-linkprop -p iov vfvnic1
LINK      PROPERTY   PERM   VALUE    EFFECTIVE   DEFAULT    POSSIBLE
vfvnic1   iov        r-     inherit  on          inherit    inherit,on,off
# dladm create-vnic -p iov=off -l net0 vnic1
# dladm show-linkprop -p iov vnic1
LINK    PROPERTY    PERM   VALUE    EFFECTIVE   DEFAULT     POSSIBLE
vnic1   iov         r-     off      off         inherit     inherit,on,off
```

この例では次の情報を指定します。

- VF VNIC を作成する前に、データリンク `net0` の `iov` プロパティーを `on` に設定する必要があります。
- VNIC の作成時に `iov` プロパティーに値を指定しないと、デフォルト値 `inherit` が `iov` プロパティーに割り当てられます。VF を持つ VF VNIC `vfvnic1` が作成されます。
- VNIC の作成時に `iov` プロパティーに値 `off` を明示的に指定すると、ベースとなるデータリンク `net0` の `iov` プロパティーが `on` の場合でも、VF を持たない通常の VNIC が作成されます。VF を持たない VNIC `vnic1` が作成されます。

VF VNIC の管理

あるデータリンクから別のデータリンクに VF VNIC または VF VLAN を移動できます。次の要件に注意してください。

- ターゲットデータリンクは SR-IOV をサポートする必要があり、`iov` プロパティーは `on` に設定する必要があります。データリンクの `iov` プロパティーのステータスを確認する方法についての詳細は、53 ページの「データリンクの SR-IOV モードの有効化」を参照してください。
- ターゲットデータリンク上で VF が使用可能である必要があります。データリンク上で使用可能な VF の数を確認する方法についての詳細は、57 ページの「VF 情報の表示」を参照してください。

これらの要件が満たされていない場合、VF VNIC は、VF を持たない通常の VNIC としてターゲットデータリンクに移行されます。

`iov=inherit` を指定することで作成された VF VNIC を移行する場合は、ターゲットデータリンクが `iov` プロパティーをサポートしていない場合や、`iov` プロパティーが無効になっている場合でも、移行は成功します。`iov=on` で作成された VF VNIC の移行を試みた場合は、ターゲットデータリンク上で SR-IOV モードが有効になっている場合のみ、移行は成功します。

VNIC を移行する方法の詳細は、48 ページの「VNIC の移行」を参照してください。

VF 情報の表示

次のコマンドを使用すると、データリンク上の VF の使用可能性に関する情報を表示できます。

```
# dladm show-phys -V
```

出力には次の情報が表示されます。

LINK	データリンクの名前。
VFS-AVAIL	VNIC に割り当てることができる、データリンク上で使用可能な VF の数。データリンクが SR-IOV をサポートしていない場合は、VFS-AVAIL が -- と表示されます。
VFS-INUSE	データリンクで使用される VF の数。データリンクが SR-IOV をサポートしていない場合は、VFS-INUSE が -- と表示されます。
FLAGS	l フラグは、データリンクが Oracle VM Server for SPARC によって管理されることを示します。

例 2-25 データリンクの VF 情報の表示

```
# dladm show-phys -V
LINK       VFS-AVAIL      VFS-INUSE      FLAGS
net0       30             1              -----
net1       0              0              l----
net2       --             --             -----
```

この例では、データリンク net0 に 30 個の使用可能な VF と 1 つの使用中の VF があります。データリンク net1 は使用可能な VF がゼロ (0) で、現在は Oracle VM Server for SPARC によって使用されています。データリンク net2 は SR-IOV をサポートしていません。

次のコマンドを使用すると、システム上の VNIC に割り当てられた VF デバイスを表示できます。

```
# dladm show-vnic -V
```

出力には次の情報が表示されます。

LINK	VNIC の名前。

`VF-ASSIGNED`	VNIC に割り当てられた VF デバイス。VNIC に VF がない場合、`VF-ASSIGNED` は `--` と表示されます。

例 2-26 VNIC に割り当てられた VF デバイスの表示

```
# dladm show-vnic -V
LINK         VF-ASSIGNED
vnic1        ixgbevf0
vnic2        --
vnic3        ixgbevf1
```

この例では、VF デバイス `ixgbevf0` が `vnic1` に割り当てられています。VNIC `vnic2` には VF デバイスは割り当てられていません。VF デバイス `ixgbevf1` は `vnic3` に割り当てられています。

♦ ♦ ♦ 第 3 章

仮想拡張ローカルエリアネットワークを使用することによる仮想ネットワークの構成

仮想ローカルエリアネットワーク (VLAN) などの従来のネットワーク分離方法は、大規模なデータセンターでの仮想化をサポートするには十分ではありません。また、クラウド環境はベースとなる物理ネットワークと密接に関連しているため、さまざまな物理レイヤー 2 ネットワークに属する物理サーバーの間で仮想マシンを移行することができません。Oracle Solaris は、大規模な仮想化データセンターやクラウド環境でのそのような仮想化の問題に対処する、仮想拡張ローカルエリアネットワーク (VXLAN) テクノロジをサポートしています。

この章では、VXLAN の配備の概要と、それらを構成する方法を説明します。また、VXLAN をゾーンなどのその他のテクノロジとともに使用する方法についても説明します。

この章の内容は、次のとおりです。

- 60 ページの「VXLAN の概要」
- 60 ページの「VXLAN を使用することの利点」
- 61 ページの「VXLAN の命名規則」
- 62 ページの「VXLAN トポロジ」
- 64 ページの「ゾーンとの VXLAN の使用」
- 67 ページの「VXLAN の構成」
- 72 ページの「VXLAN 情報の表示」
- 72 ページの「VXLAN の削除」
- 72 ページの「ゾーンへの VXLAN の割当て」
- 74 ページの「ユースケース: リンクアグリゲーション上に VXLAN の構成」

VXLAN の概要

クラウド環境では、物理サーバーがさまざまなレイヤー 2 ネットワークに配置されている場合があります。たとえば、クラウドはさまざまな地理的な場所に存在する物理サーバーにまたがる場合があります。そのような場合は、レイヤー 2 ネットワーク上に仮想マシン (VM) または*テナント*を作成すると、これらの VM のプロビジョニングに使用できる物理サーバーの数が制限されます。VM のプロビジョニングには、異なるレイヤー 2 ネットワーク上の物理サーバーを使用できます。ただし、異なるサーバー間での移行は同じレイヤー 2 ネットワークに限定されるため、物理リソースの使用は最適化されません。

VXLAN は、レイヤー 3 ネットワークの最上位にレイヤー 2 ネットワークを作成し、それによってネットワークをさらに分離するための、レイヤー 2 テクノロジです。VXLAN は、複数の物理レイヤー 2 ネットワークに広がる、仮想レイヤー 2 ネットワークを提供します。したがって、クラウド環境でのリソースのプロビジョニングは、単一の物理レイヤー 2 ネットワークに限定されません。物理サーバーが IPv4 または IPv6 ネットワークで接続されていれば、物理サーバーを VXLAN ネットワークの一部にすることができます。

VXLAN テクノロジを Oracle Solaris のエラスティック仮想スイッチ機能 (EVS) とともに使用することで、多数の仮想ネットワークを作成できます。EVS 機能を持つ VXLAN を使用して仮想ネットワークを作成する方法の詳細は、161 ページの「ユースケース: テナント用のエラスティック仮想スイッチの構成」を参照してください。詳細は、第5章「エラスティック仮想スイッチについて」および第6章「エラスティック仮想スイッチの管理」を参照してください。

VXLAN は、VXLAN セグメント ID または VXLAN ネットワーク識別子 (VNI) で識別される、分離されたレイヤー 2 セグメントを提供します。同じ VXLAN セグメント内のすべての VM は、同じ仮想レイヤー 2 ブロードキャストドメインに属します。

VXLAN での通信は、分離された VLAN での通信に似ています。したがって、同じ VXLAN セグメント内の VM だけが互いに通信できます。同じ VXLAN セグメント内にない VM は、互いに通信することはできません。

VXLAN を使用することの利点

VXLAN には次の利点があります。

- VXLAN ID は 24 ビットのため、仮想化されたクラウド環境のスケーラビリティーが向上し、最大 1600 万の分離されたネットワークを作成できます。これにより、VLAN ID が 12

ビットという VLAN の制限が取り除かれ、最大 4094 の分離されたネットワークを作成できるようになります。

- ベースとなるネットワークのレイヤー 3 機能を使用できます。
- 仮想レイヤー 2 ネットワークはベースとなる物理ネットワークから抽象化されます。その結果、仮想ネットワークは物理ネットワークからは見えず、次の利点があります。
  - 追加の物理インフラストラクチャーの必要がなくなります。たとえば、外部スイッチの転送テーブルは、サーバー上の物理ポートの背後にある VM で増加しても、増加しません。
  - MAC アドレスの重複の範囲が、同じ VXLAN セグメントに存在する VM まで減ります。MAC アドレスは、アドレスが同じ VXLAN セグメントの一部でない場合は重複する可能性があります。

    VXLAN では、同じ VXLAN セグメントまたは VNI に属するデータリンクの MAC アドレスだけが、一意である必要があります。これは、VLAN ID と MAC アドレスが一意の組み合わせを持つ必要がある VLAN に似ています。

VXLAN の命名規則

Oracle Solaris では、VXLAN エンドポイントは VXLAN データリンクによって表されます。この VXLAN データリンクは、IP アドレス (IPv4 または IPv6) と VXLAN ネットワーク識別子 (VNI) に関連付けられます。複数の VXLAN データリンクが同じ IP アドレスを使用できる場合でも、IP アドレスと VNI の組み合わせは一意である必要があります。オプションのマルチキャストアドレスを持つ VXLAN データリンクを構成して、同じ VNI でピア VXLAN エンドポイントを検出するために使用したり、VXLAN セグメント内にブロードキャストを実装するため使用することもできます。同じ VNI の VXLAN データリンクには、同じマルチキャストアドレスを構成する必要があります。VXLAN の要件についての詳細は、66 ページの「VXLAN の要件」を参照してください。

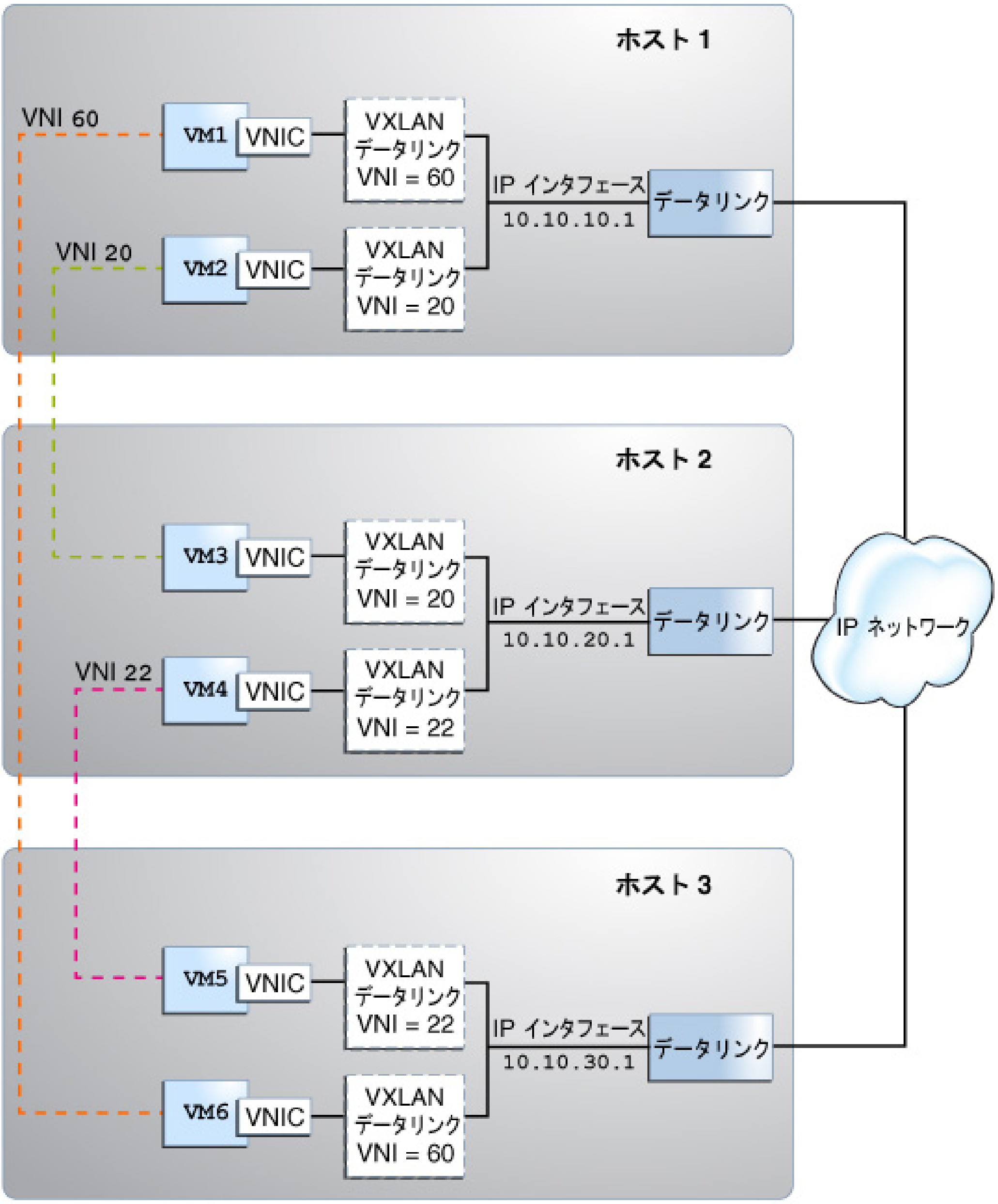
すべての VXLAN データリンクは、VXLAN セグメント ID、つまり VNI に関連付けられます。VXLAN データリンクの命名規則は、リンクまたは VLAN に使用される規則と同じです。有効なデータリンク名の指定についての詳細は、『Oracle Solaris 11.2 でのネットワークコンポーネントの構成と管理 』の「有効なリンク名のための規則」を参照してください。

VXLAN トポロジ

VXLAN を使用すると、レイヤー 3 ネットワーク上のシステムを、独自の VXLAN セグメント内に構成できます。

次の図は、複数の物理サーバー上に構成されている VXLAN ネットワークを示しています。

図 3-1 VXLAN トポロジ



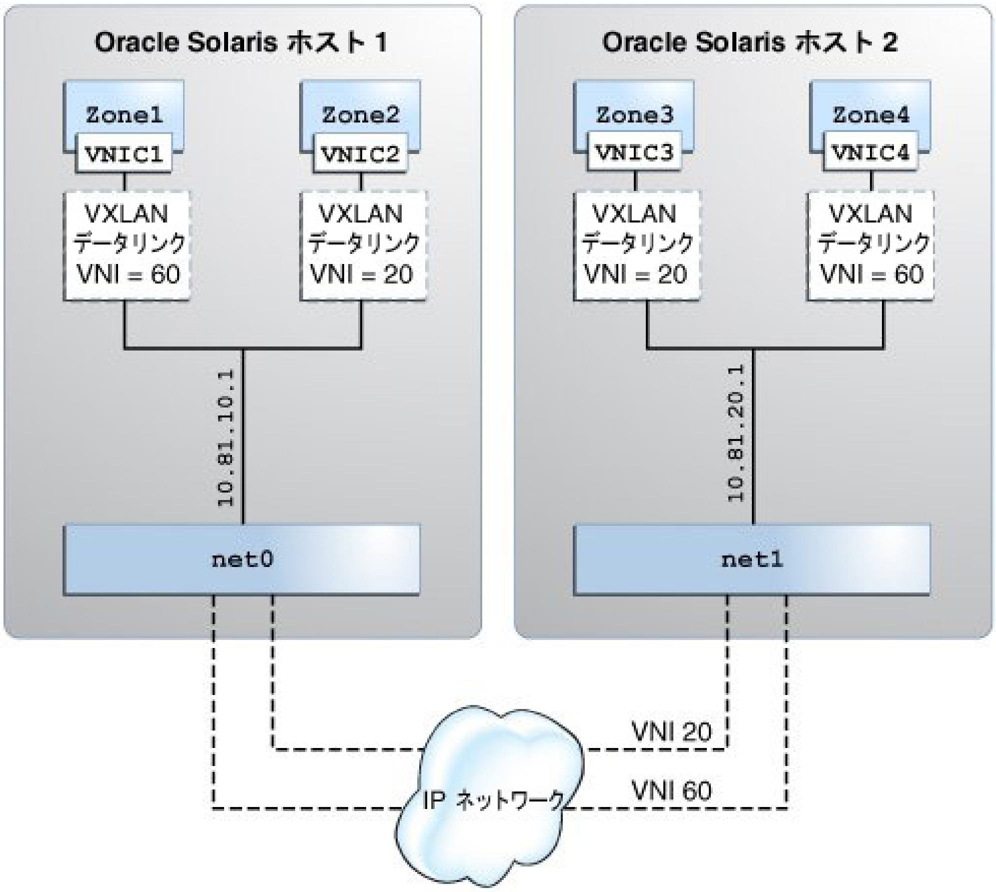
この図は、IP ネットワークインフラストラクチャーに接続された 3 つの仮想化されたホストを示しています。VXLAN セグメント ID、つまり VNI 60、20、および 22 で識別される 3 つの VXLAN オーバーレイネットワークがあります。VM VM1 および VM6 は VNI 60 で識別されるオーバーレイネットワーク上にあり、VM VM2 および VM3 は VNI 20 で識別されるオーバーレイネットワーク上にあり、VM VM4 および VM5 は VNI 22 で識別されるオーバーレイネットワーク上にあります。

ゾーンとの VXLAN の使用

VXLAN データリンク上に作成された VNIC を、ゾーンに割り当てることができます。VXLAN データリンクは VNI を指定することによって作成され、これらの VXLAN データリンクはその VNI によって識別される VXLAN セグメントに属します。たとえば、VXLAN データリンクを作成するときに VNI を 20 として指定した場合、そのデータリンクは VNI 20 で識別される VXLAN セグメントに属します。VXLAN データリンク上に作成される VNIC は、VXLAN セグメントの一部です。

次の図は、VNI 20 および 60 で識別される 2 つの VXLAN オーバーレイネットワークがある、IP ネットワークインフラストラクチャーに接続された 2 つの仮想化された Oracle Solaris ホストを示しています。

図 3-2 VXLAN とゾーン



VXLAN セグメントの一部であるゾーンは、次の方法で作成できます。

- VXLAN 上に VNIC を作成し、その VNIC をゾーンに割り当てます。詳細は、67 ページの「VXLAN の構成」を参照してください。
- ゾーンの anet (VNIC) リソースのベースとなるリンクとして、VXLAN を割り当てます。詳細は、72 ページの「ゾーンへの VXLAN の割当て」を参照してください。

どのような場合でも、ゾーン内に作成された VNIC は、ベースとなる VXLAN データリンクで識別される VXLAN セグメントの一部です。ゾーンについての詳細は、『Oracle Solaris 11.2 仮想化環境の紹介 』を参照してください。

VXLAN リンクへの VNIC の割り当ては、VLAN リンクを作成してゾーンに割り当てることと同じです。VLAN の作成とゾーンへの割り当てについての詳細は、『Oracle Solaris 11.2 でのネットワークデータリンクの管理 』の「VLAN を構成する方法」を参照してください。

VXLAN 構成の計画

VXLAN 構成の計画には、次の手順が含まれます。

1. 物理ネットワークの仮想ネットワークトポロジを決定します。たとえば、さまざまなサーバー上の複数の VM で構成されるサービスをホストする場合は、これらの VM に VXLAN セグメントを割り当てることができます。この VXLAN セグメント内の VM は互いに通信できますが、この VXLAN セグメント内にないほかの VM とは通信できません。
2. 物理サーバーが IP インタフェースを介して接続されていること、および IP マルチキャストが物理ネットワーク上で有効になっていることを確認します。
3. VXLAN セグメントの番号付けスキームを作成します。たとえば、VM によってホストされるアプリケーションに基づいて、VXLAN セグメント (VNI) を割り当てることができます。
4. IP アドレスと VXLAN セグメント ID を指定することによって、VXLAN データリンクを作成します。

   オプションで、独自のマルチキャストアドレスを持つ VXLAN セグメントを割り当てることができます。
5. VXLAN データリンク上に VNIC を作成し、その VNIC をゾーンに割り当てます。

   または、ゾーンの `anet` リンクのベースとなるリンクとして、VXLAN リンクを割り当てることができます。

VXLAN の要件

VXLAN を使用する前に、次の要件を満たしているかどうかを確認してください。

- ネットワークで IP マルチキャストがサポートされていることを確認してください。IP マルチキャストがサポートされていない場合、VXLAN 内の VM は互いに通信することができません。
- VXLAN にさまざまな IP サブネットのサーバーが含まれている場合は、サブネット全体でマルチキャストルーティングがサポートされている必要があります。マルチキャストルーティングがサポートされていない場合は、同じ IP サブネットの VXLAN 上の VM だけが互いに通

信でき、別の IP サブネットの VXLAN 上の VM (たとえば、地理的に分散されたデータセンター) は、互いに通信することはできません。

VXLAN データリンクの命名規則についての詳細は、61 ページの「VXLAN の命名規則」を参照してください。

VXLAN の構成

次の手順では、ゾーンがすでにシステム上に作成されていることを前提としています。ゾーン構成の詳細は、『Oracle Solaris ゾーンの作成と使用 』の第 1 章「非大域ゾーンの計画および構成方法」を参照してください。

▼ VXLAN を構成する方法

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **システム上の使用可能な IP アドレスを確認します。**

   ```
   # ipadm show-addr
   ```

3. **IP アドレスまたは IP インタフェースを指定することによって、VXLAN データリンクを作成します。**

   - **IP アドレスを指定することによって VXLAN を作成するには:**

     ```
     # dladm create-vxlan -p prop=value VXLAN-LINK
     ```

     -p *prop=value* 　作成する VXLAN データリンクの指定された値に設定できる、コンマ区切りの VXLAN データリンクプロパティーのリストを指定します。次のプロパティーを設定します。

     - `addr` - VXLAN ネットワークの IPv4 または IPv6 アドレスを指定します。このアドレスは、特定のアドレスまたはアドレス/接頭辞の長さの組み合わせを指定できます。
     - `vni` - VXLAN セグメントのネットワーク識別子を指定します。0 から 16777215 まで数を指定できます。

- `mgroup` - (オプション) マルチキャストグループ名を指定します。このオプションは、VXLAN セグメントに独自のマルチキャストグループがある場合にのみ指定できます。

VXLAN-LINK VXLAN の名前。

- **IP インタフェースを指定することによって VXLAN を作成するには:**

  # dladm create-vxlan -p *prop=value*

  `-p` *prop=value* 作成する VXLAN データリンクの指定された値に設定できる、コンマ区切りの VXLAN データリンクプロパティーのリストを指定します。次のプロパティーを設定します。

  - `interface` – VXLAN ネットワーク用の IP インタフェースを指定します。
  - `vni` - VXLAN セグメントのネットワーク識別子を指定します。0 から 16777215 まで数を指定できます。

  VXLAN VXLAN の名前。

  IP インタフェースと IP バージョンを指定した場合、VXLAN データリンクは、そのインタフェースで指定されているバージョンの使用可能な IP アドレス上に作成されます。たとえば、`net0` 上に IP アドレス **10.10.10.1** が構成されている場合、VXLAN データリンクは **10.10.10.1** 上に作成されます。デフォルトでは、IP バージョンは IPv4 アドレスです。ただし、IPv6 アドレスが必要な場合は、`ipvers` プロパティーを使用してバージョンを指定する必要があります。

注記 - VXLAN データリンクは、物理集約リンク (トランクまたは DLMP アグリゲーション) または IPoIB リンク上でホストされている IP アドレス上に作成できます。ただし、IPMP、仮想ネットワークインタフェース、またはループバックインタフェースでホストされている IP アドレスに VXLAN データリンクを作成することはできません。

---

4. **作成した VXLAN を確認します。**

   # dladm show-vxlan

5. **VXLAN データリンク上に VNIC を作成します。**

   # dladm create-vnic -l *VXLAN-LINK VNIC*

VXLAN データリンク上に VLAN VNIC を作成できます。VLAN VNIC を作成するには、-f (force) オプションを指定する必要があります。詳細は、28 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。

6. **VNIC 上に直接、または VNIC を最初にゾーンに割り当てることによって、IP インタフェースを構成します。**

   - **VNIC 上に IP インタフェースを構成します。**

     ```
     # ipadm create-ip VNIC
     ```

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

   - **VNIC をゾーンに割り当てて、ゾーン内の VNIC 上に IP インタフェースを構成します。**

     a. **ゾーンのインタフェースで VNIC を割り当てます。**

     ```
     zonecfg:zone> add net
     zonecfg:zone:net> set physical=VNIC
     zonecfg:zone:net> end
     ```

     b. **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:zone> verify
     zonecfg:zone> commit
     zonecfg:zone> exit
     ```

     c. **ゾーンをリブートします。**

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

     d. **ゾーンにログインします。**

     ```
     global# zlogin zone
     ```

     e. **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。**

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

     f. **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
```

-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。

g. ゾーンを終了します。

dladm および ipadm コマンドの詳細は、dladm(1M) および ipadm(1M) のマニュアルページを参照してください。

例 3-1 VXLAN の作成と、VXLAN 上に作成された VNIC 用の IP インタフェースの構成

1. システム上の使用可能な IP アドレスを確認します。

```
# ipadm show-addr net4
ADDROBJ   TYPE   STATE   ADDR
net4/v4   static ok      10.10.11.1/24
```

2. VXLAN セグメント 10 に VXLAN データリンクを作成します。

```
# dladm create-vxlan -p addr=10.10.11.1,vni=10 vxlan1
```

3. 作成した VXLAN リンクを確認します。

```
# dladm show-vxlan
LINK   ADDR       VNI  MGROUP
vxlan1 10.10.11.1 10   224.0.0.1
```

マルチキャストアドレスが指定されていないため、この VXLAN セグメントは、同じネットワークセグメント上のすべてのホストに対応する All Host マルチキャストアドレスを使用します。

4. VXLAN リンク情報を確認します。

```
# dladm show-link vxlan1
LINK   CLASS MTU  STATE OVER
vxlan1 vxlan 1440 up    --
```

vxlan1 が作成され、リンク状態は up です。

5. vxlan1 上に VNIC を作成します。

```
# dladm create-vnic -l vxlan1 vnic1
```

6. 作成した VNIC を確認します。

```
# dladm show-vnic
LINK    OVER    SPEED  MACADDRESS       MACADDRTYPE  VIDS
vnic1   vxlan1 10000   2:8:20:fe:58:d4 random       0
```

7. VNIC 上に IP インタフェースを構成します。

```
# ipadm create-ip vnic1

# ipadm create-addr -T static -a local=10.10.12.1/24 vnic1/v4

# ipadm show-addr vnic1
ADDROBJ   TYPE    STATE  ADDR
vnic1/v4 static ok      10.10.12.1/24
```

VXLAN は IP アドレスを指定することによって正常に作成されました。VXLAN 上に VNIC が作成され、IP インタフェースが構成されました。

例 3-2 VXLAN 上に作成された VNIC のゾーンへの割り当てと IP インタフェースの構成

この例は、例3-1「VXLAN の作成と、VXLAN 上に作成された VNIC 用の IP インタフェースの構成」の手順 1 から 6 を完了済みであることを前提としています。

VNIC を作成したあと、VNIC をゾーンに割り当てて、IP インタフェースを構成します。

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

global# zlogin zone2
zone2# ipadm create-ip vnic1
zone2# ipadm create-addr -a 192.168.3.85/24 vnic1
ipadm: vnic1/v4

zone2# exit
```

VNIC がゾーンに割り当てられ、VNIC 上に IP インタフェースが構成されました。

VXLAN 情報の表示

dladm show-link コマンドを使用すると、VXLAN リンクに関する一般的なリンク情報を表示できます。VXLAN に固有の情報を表示するには、dladm show-vxlan コマンドを使用します。

```
# dladm show-link
LINK                CLASS     MTU    STATE    OVER
net6                phys      1500   down     --
net0                phys      1500   up       --
net2                phys      1500   unknown  --
net3                phys      1500   unknown  --
net1                phys      1500   unknown  --
net5                phys      1500   unknown  --
net4                phys      1500   up       --
vxlan1              vxlan     1440   up       --
vnci1               vnic      1440   up       vxlan1

# dladm show-vxlan vxlan1
LINK           ADDR           VNI   MGROUP
vxlan1         10.10.11.1     10    224.0.0.1
```

VXLAN の削除

VXLAN リンクを削除するには、dladm delete-vxlan コマンドを使用します。VXLAN リンクを削除する前に、dladm show-link コマンドを使用して、その VXLAN リンク上に VNIC が構成されていないことを確認する必要があります。

管理者になり、次のコマンドを発行します。

 # dladm delete-vxlan *VXLAN*

たとえば、vxlan1 を削除する場合は、次のコマンドを入力します。

```
# dladm delete-vxlan vxlan1
```

ゾーンへの VXLAN の割当て

VXLAN をゾーンの anet リソースへのベースとなるリンクとして割り当てることで、VXLAN セグメントの一部であるゾーンを作成できます。ゾーンの構成についての詳細は、『Oracle Solaris ゾーンの作成と使用 』を参照してください。

▼ ゾーンに VXLAN を割り当てる方法

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **システム上の使用可能な IP アドレスを確認します。**

   ```
   # ipadm show-addr
   ```

3. **IP アドレスを指定することによって VXLAN を作成します。**

   ```
   # dladm create-vxlan -p prop=value VXLAN-LINK
   ```

4. **作成した VXLAN を確認します。**

   ```
   # dladm show-vxlan
   ```

5. **ゾーンの anet のベースとなるリンクとして作成した VXLAN を割り当てることによって、ゾーンを構成します。**

   ```
   global# zonecfg -z zone
   zonecfg:zone2> add anet
   zonecfg:zone2:net> set linkname=datalink
   zonecfg:zone2:net> set lower-link=VXLAN-LINK
   zonecfg:zone2:net> end
   zonecfg:zone2> verify
   zonecfg:zone2> commit
   zonecfg:zone2> exit
   global# zoneadm -z zone reboot
   ```

   VXLAN が、ゾーンの anet のベースとなるリンクとして割り当てられます。

例 3-3 ゾーンの anet への VXLAN の割り当て

```
# ipadm show-addr net4
ADDROBJ   TYPE   STATE   ADDR
net4/v4   static ok      10.10.11.1/24 2

# dladm create-vxlan -p addr=10.10.11.1,vni=10 vxlan1

# dladm show-vxlan
LINK   ADDR       VNI  MGROUP
vxlan1 10.10.11.1 10   224.0.0.1
```

マルチキャストアドレスが指定されていないため、この VXLAN セグメントは、同じネットワークセグメント上のすべてのホストに対応する All Host マルチキャストアドレスを使用します。

```
# dladm show-link vxlan1
LINK   CLASS MTU  STATE OVER
vxlan1 vxlan 1440 up    --
```

vxlan1 が作成され、リンク状態は up です。

```
global# zonecfg -z zone2
zonecfg:zone2> add anet
zonecfg:zone2:net> set linkname=net1
zonecfg:zone2:net> set lower-link=vxlan1
zonecfg:zone2:net> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> exit
global# zoneadm -z zone2 reboot
```

vxlan1 が、ゾーンの anet のベースとなるリンクとして割り当てられます。

ゾーンがブートすると、net1 が vxlan1 上の zone2 内に作成されます。

ユースケース: リンクアグリゲーション上に VXLAN の構成

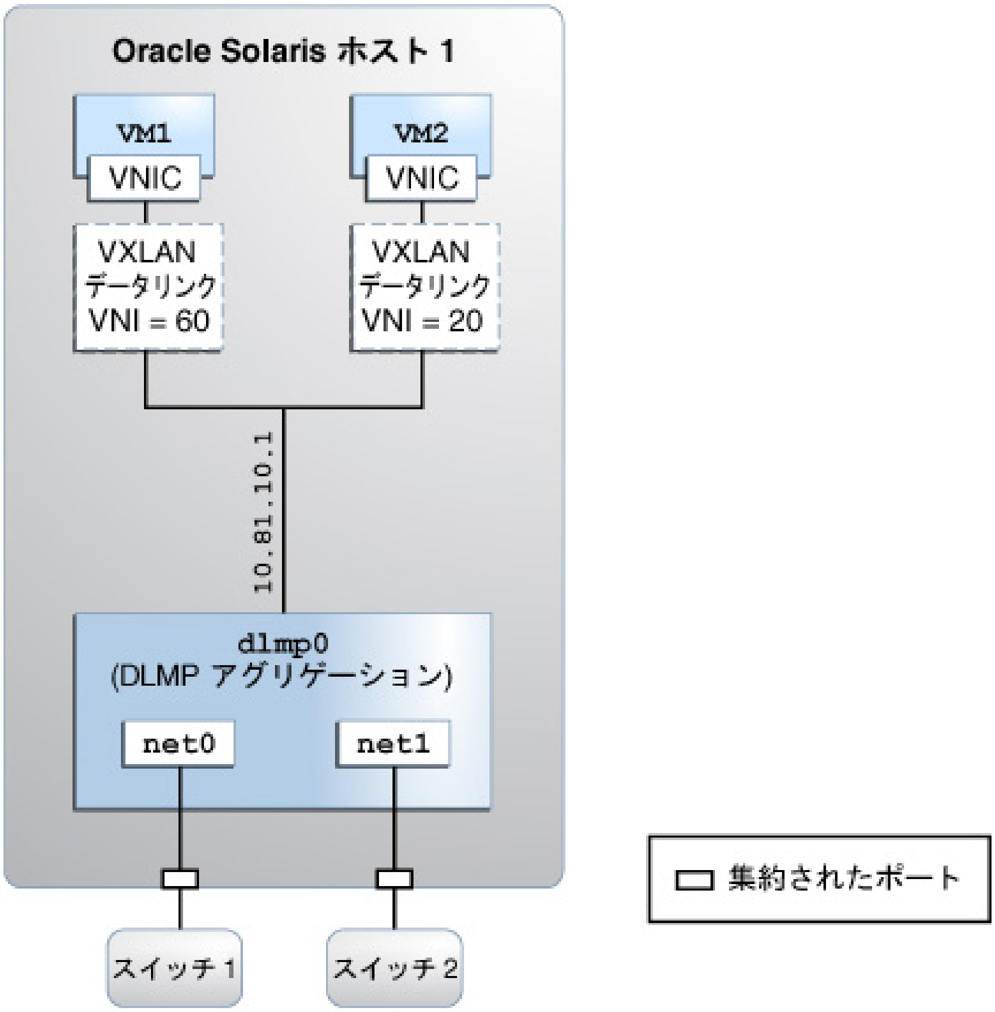
次のユースケースでは、次のことを実行する方法を示します。

- DLMP アグリゲーションを作成する
- アグリゲーション上に IP アドレスを構成する
- アグリゲーション上に 2 つの VXLAN を作成する
- 下位リンクとして VXLAN データリンクがある 2 つのゾーンの構成

リンクアグリゲーションの詳細は、『Oracle Solaris 11.2 でのネットワークデータリンクの管理 』の第 2 章「リンクアグリゲーションを使用した高可用性の構成」を参照してください。

次の図は、DLMP アグリゲーション上の VXLAN 構成を示しています。

図 3-3 リンクアグリゲーション上の VXLAN



集約されたポートや外部スイッチに障害が発生しても、アグリゲーション上の VXLAN データリンクは少なくとも 1 つのポートとスイッチが機能していれば引き続き存在するため、フェイルオーバー中のネットワーク高可用性が提供されます。たとえば、`net0` に障害が発生した場合、DLMP アグリゲーションは、残りのポート `net1` を VXLAN データリンク間で共有します。集約されたポート間の分散は、ユーザーに対しては透過的に、また、アグリゲーションに接続されている外部スイッチとは無関係に行われます。

1. 管理者になります。

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. アグリゲーション用のデータリンクを識別するため、データリンク情報を表示します。

```
# dladm show-link
LINK      CLASS     MTU     STATE   OVER
net0      phys      1500    up      --
net1      phys      1500    up      --
net2      phys      1500    up      --
```

3. 集約するデータリンクのリンク上に IP インタフェースが構成されていないことを確認します。インタフェースがリンクのいずれかに構成されている場合は、そのインタフェースを削除します。

```
# ipadm show-if
IFNAME      CLASS       STATE     ACTIVE     OVER
lo0         loopback    ok        yes        --
net0        ip          ok        no         --
# ipadm delete-ip net0
```

4. リンク net0 および net1 のある DLMP アグリゲーションを作成します。

```
# dladm create-aggr -m dlmp -l net0 -l net1 dlmp0
```

5. アグリゲーション dlmp0 の最上位に IP インタフェースを構成します。

```
# ipadm create-ip dlmp0
# ipadm create-addr -T static -a local=10.10.10.1 dlmp0/v4
```

6. アグリゲーション上に構成されている IP アドレスを指定して 2 つの VXLAN を作成し、VNI (VXLAN セグメントのネットワーク識別子) も指定します。

```
# dladm create-vxlan -p addr=10.10.10.1,vni=20 vxlan20
# dladm create-vxlan -p addr=10.10.10.1,vni=60 vxlan60
```

   両方の VNI にデフォルトのマルチキャストアドレスが構成されます。

7. ゾーン VM1 に、下位リンクとして VXLAN データリンク vxlan20 を構成します。

```
global# zonecfg -z VM1
zonecfg:VM1> add anet
zonecfg:VM1:net> set linkname=net0
zonecfg:VM1:net> set lower-link=vxlan20
```

```
zonecfg:VM1:net> end
zonecfg:VM1> verify
zonecfg:VM1> commit
zonecfg:VM1> exit
global# zoneadm -z VM1 reboot
```

8. ゾーン `VM2` に、下位リンクとして VXLAN データリンク `vxlan60` を構成します。

```
global# zonecfg -z VM2
zonecfg:VM2> add anet
zonecfg:VM2:net> set linkname=net0
zonecfg:VM2:net> set lower-link=vxlan60
zonecfg:VM2:net> end
zonecfg:VM2> verify
zonecfg:VM2> commit
zonecfg:VM2> exit
global# zoneadm -z VM2 reboot
```

`net0` データリンクと `net1` データリンクが DLMP アグリゲーション内に集約され、`dlmp0` と IP アドレス `10.10.10.1` がアグリゲーション用に構成されます。VXLAN、`vxlan20` および `vxlan60` が、アグリゲーション用に構成された、指定した IP アドレス `10.10.10.1` 上に作成されます。VXLAN `vxlan20` は VXLAN セグメント `20` に作成され、VXLAN `vxlan60` は VXLAN セグメント `60` に作成されます。ゾーン `VM1` に下位リンクとして VXLAN データリンク `vxlan20` が構成され、ゾーン `VM2` に下位リンクとして VXLAN データリンク `vxlan60` が構成されます。

♦ ♦ ♦ 第 4 章

エッジ仮想ブリッジングを使用したサーバーネットワークエッジの仮想化の管理

サーバーネットワークエッジは、サーバーポートとその最初のホップスイッチポートの間の接続に存在します。仮想ローカルエリアネットワーク (VLAN) や LACP (Link Aggregation Control Protocol) などのネットワーク構成は、サーバーポートとスイッチポート上のこのエッジで同じである必要があります。データセンターブリッジング機能交換 (DCBX) を使用すると、サーバーとスイッチポート上の構成を自動化できます。詳細は、『Oracle Solaris 11.2 でのネットワークデータリンクの管理 』の第 6 章「データセンターブリッジングを使用した集中ネットワークの管理」を参照してください。

サーバーの仮想化では、スイッチポートに接続された 1 つだけのサーバーポートではなく、複数の仮想ポートがサーバーポートの背後にある仮想マシン (VM) に関連付けられます。サーバーの仮想化では、サーバーネットワークエッジに次の追加の要件が課せられます。

- VM 間のトラフィックがスイッチ上で構成されたポリシーに従うように、外部スイッチ経由での仮想マシンの切り替え
- 仮想ポートのプロパティーのネットワークへの拡張

Oracle Solaris は、これらの要件に対応する発展性のある IEEE 標準であるエッジ仮想ブリッジング (EVB) をサポートしています。

この章の内容は、次のとおりです。

- 80 ページの「サーバーネットワークエッジの仮想化における EVB のサポート」
- 81 ページの「Improving Network and Server Efficiency by Using EVB」
- 84 ページの「EVB のインストール」
- 85 ページの「同じ物理ポート上の VM 間の切り替えの制御」
- 90 ページの「VDP を使用した VNIC 情報の交換」
- 92 ページの「VDP と ECP の状態と統計情報」
- 94 ページの「デフォルトの EVB 構成の変更」

サーバーネットワークエッジの仮想化における EVB のサポート

仮想化されたサーバーでは、同じ物理リンク上に複数の仮想 NIC が含まれている可能性があります。これらの VNIC を VM に割り当てることができます。従来より、スイッチが、パケットを受信したのと同じリンク上でパケットを送り返すことはありません。VM 間のパケットは、ホスト自体の内部にある仮想スイッチによってループバックされます。そのため、外部スイッチ上で構成されているどのポリシーも VM 間のパケットには適用されません。EVB をサポートしている場合は、Oracle Solaris とスイッチにより、VM 間のパケットにすべてのポリシーを適用したあと、外部スイッチで VM 間のパケットを切り替えることができるようになります。VNIC の詳細は、第2章「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。

さらに、EVB をサポートしている Oracle Solaris は、VNIC に関する情報をスイッチと交換できます。この情報の交換により、スイッチは、ネットワーク上の帯域幅制限、帯域幅共有、MTU などの VNIC プロパティーを自動的に構成できるようになります。この機能がない場合、サーバー上で VNIC が作成、変更、または削除されるたびに、サーバー管理者とネットワーク管理者はお互いに調整してスイッチ上の変更を行う必要があります。VNIC プロパティーをネットワークに拡張することによって、VNIC プロパティーに基づいたネットワークリソースの効率的な使用が可能になります。たとえば、ホストに到着したあとのパケットに帯域幅制限を適用しても、それらのパケットがすでにリンクの帯域幅を使い果たしている可能性があるため、あまり役に立ちません。

反射型リレー

反射型リレーは、同じ物理 NIC 上で VNIC を使用している複数の VM が外部スイッチ経由で通信できるようにする機能です。スイッチがこの機能をサポートしている必要があります。Oracle Solaris では、LLDP が拡張され、スイッチが反射型リレー機能をサポートしているかどうかを判定したり、スイッチ上の反射型リレー機能を有効または無効にしたりするために使用される EVB Type-Length Value (TLV) ユニットが含まれます。そのため、LLDP を使用してスイッチ上でのこの機能の検出や構成を自動化できるのは、スイッチが LLDP および EVB TLV ユニットをサポートしている場合だけです。それ以外の場合は、スイッチ上で反射型リレー機能を手動で構成する必要があります。反射型リレーを手動で構成する方法については、スイッチの製造元のドキュメントを参照してください。

Oracle Solaris での反射型リレーのサポートの詳細は、85 ページの「同じ物理ポート上の VM 間の切り替えの制御」を参照してください。LLDP の TLV ユニットについての詳細

は、『Oracle Solaris 11.2 でのネットワークデータリンクの管理 』の「LLDP エージェントが通知する情報」を参照してください。

ネットワーク内の VNIC の自動構成

Oracle Solaris は、IEEE 802.1Qbg で定義されている VDP (Virtual Station Interface Discovery and Configuration Protocol) を使用して VNIC 情報をスイッチと交換します。スイッチが VDP をサポートしている場合は、スイッチ上で VNIC プロパティーが自動的に構成されます。これは、DCBX を使用した、ホストとスイッチの間での物理リンクのプロパティーの交換に似ています。VNIC が作成、変更、または削除されると、ホストとスイッチの間で VDP の交換が開始されます。この交換により、スイッチは VNIC のプロパティーに基づいて、VNIC 宛てのパケットにリソースを割り当てることができるようになります。

Oracle Solaris でのシステムと外部スイッチ間での VNIC 情報についての交換の詳細は、90 ページの「VDP を使用した VNIC 情報の交換」および91 ページの「VDP による VNIC 情報の交換方法」を参照してください。

Improving Network and Server Efficiency by Using EVB

このセクションでは、サーバー上で EVB を有効にした場合に、サーバーおよびネットワークの効率をどのように向上させることができるかを示す例について説明します。

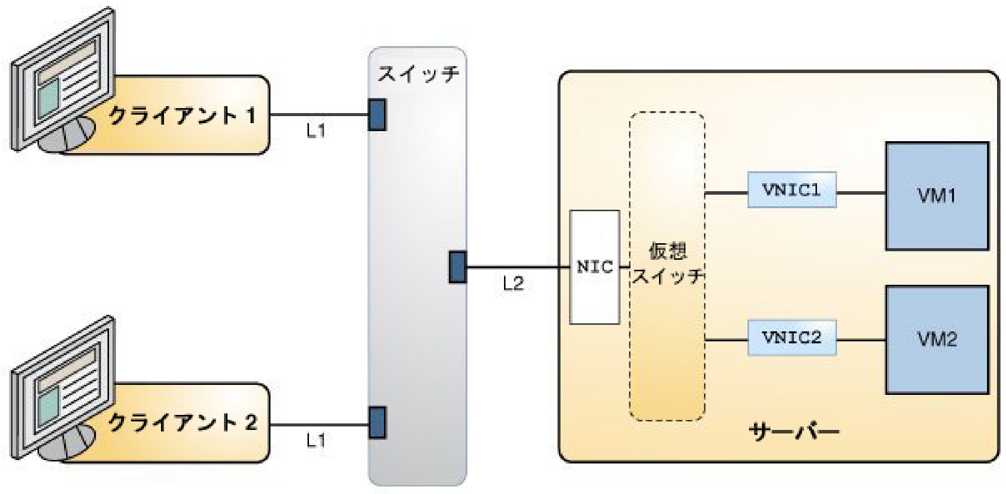
この例では、サーバーが同じ物理マシンのクラウド環境で 2 つのアプリケーションをホストしていることを前提としています。

- アプリケーションは物理マシン上で別個の仮想マシン (VM1 と VM2) としてクラウド上でホストされます。VNIC の `VNIC1` と `VNIC2` は、それぞれ VM1 と VM2 用に構成されます。
- アカウントを持つクライアント (クライアント 1 とクライアント 2) はアプリケーションにアクセスできます。
- 仮想マシン (VM1 と VM2) は、物理システムのリソースとリンク L2 上の帯域幅を共有します。
- クライアントはリンク L1 を使用してスイッチに接続されています。スイッチはリンク L2 を使用して NIC に接続されています。
- あらかじめ定義された SLA が、仮想マシンのリソースの割り当てを決定します。仮想マシンの SLA には、次 (L2) の帯域幅の使用量が含まれています。

- VM1 は優先度の高い Transmission Control Protocol (TCP) サービスを実行しています。したがって、VM1 の SLA には最大帯域幅 8G ビット/秒の制限があります。
- VM2 は、優先度の高くない User Datagram Protocol (UDP) サービスを実行しています。したがって、VM2 の SLA には最大帯域幅 3G ビット/秒の制限があります。

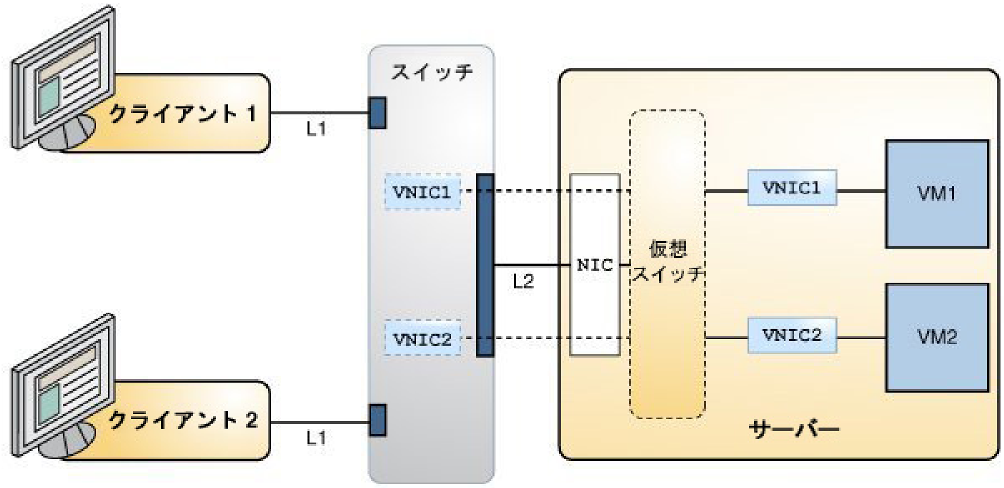
次の図は、サーバーでホストされているアプリケーションを示しています。

図 4-1 EVB を使用しない場合のアプリケーション設定



サーバーとスイッチで EVB を有効にすると、サーバーは、次の図に示すように、同じ物理スイッチポート経由でスイッチと VNIC 情報を交換します。

図 4-2　　EVB が有効になっている場合のアプリケーション設定



次の表に、サーバーとスイッチで EVB を有効にする前とあとのサーバーの効率を示します。

表 4-1　　EVB を使用する場合と EVB を使用しない場合のサーバー効率

EVB を使用しない場合のサーバーの効率	EVB を使用する場合のサーバーの効率
サーバーが帯域幅の適用のためにクライアントからの受信トラフィックを規制します。	スイッチがサーバー宛てのトラフィックを規制します。
システムリソースが使用されるため、システムとネットワークのパフォーマンスに影響します。	システムリソースが帯域幅の処理に使用されないため、システム効率が向上します。
この例では、クライアント (クライアント 1 とクライアント 2) がサービスを同時に利用する必要がある場合、各クライアントはリンク L2 の帯域幅とサーバーリソースを使用します。サーバーはクライアントのインバウンドおよびアウトバウンドのトラフィックを規制するために、VM1 と VM2 の VNIC で SLA を適用します。ただし、ネットワークのパフォーマンスと帯域幅の使用量は次の点で影響を受けます。 ■ クライアント (クライアント 1 とクライアント 2) からのトラフィックは、リンク L2 の帯域幅を無制限に使用します。また、ホスト上で帯域幅制限が構成されている場合は、L2 の帯域幅を使用するパケットがホスト上で破棄されることもあり、それによって帯域幅が効率的に使用されなくなります。	EVB がサーバーとスイッチで有効になっている場合、システムの効率は次の点で向上します。 ■ サーバーの VNIC に構成されている SLA がスイッチに反映されます。 ■ スイッチは、構成済みの帯域幅に基づいて VM1 と VM2 に向かうトラフィックを規制するため、リンク L2 の帯域幅の適切な利用に役立ち、これによってネットワークの効率が向上します。 スイッチが帯域幅を規制するため、サーバーは受信側で帯域幅を処理する必要がなくなり、これによってサーバーの効率が向上します。

EVB を使用しない場合のサーバーの効率	EVB を使用する場合のサーバーの効率
■ VM1 は優先度の高い TCP サービスを提供し、VM2 は優先度の高くない UDP サービスを提供します。サーバー上の VM1 の帯域幅を規制すると、TCP が応答するため、リンク L2 上の帯域幅の VM1 の使用に影響を与えます。ただし、サーバー上で VM2 のサービスを規制しても、リンク L2 の帯域幅の使用状況には影響を与えません。これはリンク L2 を使用しているほかのサービスに影響を与えます。	
この例では、UDP および TCP サービスのネットワークトラフィックのサーバーへのインバウンドは、リンク L2 上の使用可能な帯域幅を無制限に使用します。サーバーがネットワークトラフィックを受信したあと、構成済みの帯域幅の制限に基づいてネットワークトラフィックを規制します。	構成済みの帯域幅制限 (3G ビット/秒と 8G ビット/秒) は、サーバーに加えスイッチによって規制されます。したがって、共有リンク L2 の使用量は構成済みの帯域幅制限に基づきます。

EVB のインストール

システムで EVB を使用するには、EVB パッケージをインストールする必要があります。

▼ EVB をインストールする方法

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **EVB パッケージがインストールされているかどうかを確認します。**

   ```
   # pkg info evb
   ```

3. **EVB パッケージがインストールされていない場合は、このパッケージをインストールします。**

   ```
   # pkg install evb
   ```

4. **サービスが使用可能になっているかどうかを確認します。**

   ```
   # svcs vdp
   ```

5. **サービスが使用可能でない場合は、サービスを使用可能にします。**

   ```
   # svcadm enable vdp
   ```

デフォルトの EVB 構成は、EVB パッケージのインストール後に自動的に有効になります。デフォルトの EVB 構成を受け入れることで、システムは、システムに外部スイッチを構成する VNIC に関する情報をすぐに交換できます。

参照

- VNIC 情報の交換、VNIC 情報の交換に使用されるプロトコル、および EVB コンポーネントの詳細は、90 ページの「VDP を使用した VNIC 情報の交換」を参照してください。
- システムで EVB が有効になっている場合に物理 Ethernet リンクの VDP (VSI Discovery and Configuration Protocol) 状態に関する情報を表示し、VNIC 用に VDP パケットが交換されているかどうかも確認するには、92 ページの「VDP と ECP の状態と統計情報」を参照してください。
- デフォルトの EVB 構成を変更するには、94 ページの「デフォルトの EVB 構成の変更」を参照してください。
- デフォルトの EVB 構成情報を表示するには、例4-2「物理リンクでの EVB 関連のデータリンクプロパティーの表示例」 を参照してください。

同じ物理ポート上の VM 間の切り替えの制御

`vswitchmode` データリンクプロパティーを使用すると、同じ物理ポート上の VM の切り替えを制御できます。3 つの値を指定できます。

- `local` - 同じ物理 NIC 上の VM 間のネットワークトラフィックを内部的に交換できるようにします。これは、デフォルトのモードです。
- `remote` - 同じ物理 NIC 上の VM 間のネットワークトラフィックを外部スイッチ経由で交換できるようにします。
- `auto` - LLDP を使用して、反射型リレーが外部スイッチ上でサポートされているかどうかを確認します。反射型リレーが外部スイッチでサポートされている場合、VM 間のネットワークトラフィックは外部スイッチ経由で交換されます。それ以外の場合、VM 間のネットワークトラフィックは内部的に交換されます。

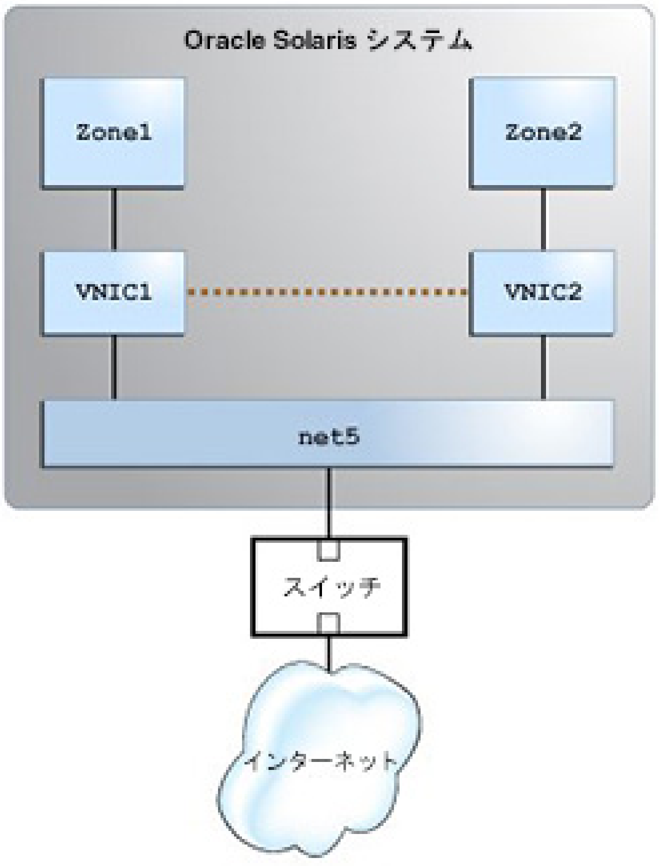
外部スイッチ経由での VM の通信の有効化

同じ物理 NIC 上に複数の VNIC が構成されている場合は、`vswitchmode` データリンクプロパティーを `remote` に設定して、ネットワークトラフィックをスイッチ経由で外部に送信できます。た

だし、外部スイッチを反射型リレーモードで構成する必要があります。反射型リレーを有効にするスイッチ構成は、スイッチのタイプに固有です。詳細は、スイッチの製造元のドキュメントを参照してください。

次の図は、外部スイッチに接続し、同じ顧客のサービスを実行している 2 つのゾーン (VM) をホストしている、10G Ethernet リンクのサンプルのシステムを示しています。

図 4-3 ゾーン間の内部通信



Zone1 と Zone2 という 2 つのゾーンが同じ顧客のサービスを実行しているため、2 つのゾーン間の通信は制限なしで内部的に発生する可能性があります。したがって、VNIC1 と VNIC2 の間のトラフィックは内部的に交換できます。

次のように、物理 NIC net5 の vswitchmode プロパティーの既存の値を確認します。

```
# dladm show-linkprop -p vswitchmode net5
LINK  PROPERTY     PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
net4  vswitchmode  rw    local  local      local    local,remote,auto
```

出力の VALUE および EFFECTIVE フィールドに値 local が表示されます。この値はゾーン間の通信が内部的であることを示しています。

この例では、Zone1 と Zone2 という 2 つのゾーンが異なる顧客向けのサービスを実行する必要があり、これらのサービスのネットワークトラフィックを制御するためのアクセス制御リスト (ACL) が外部スイッチに構成されているものとします。そのため、これらのゾーンは内部的には通信できず、VNIC1 と VNIC2 の間のネットワークトラフィックをスイッチ経由で外部的に交換する必要があります。

したがって、次のように vswitchmode プロパティーを remote に設定することによって、ゾーン間の内部通信を無効にする必要があります。

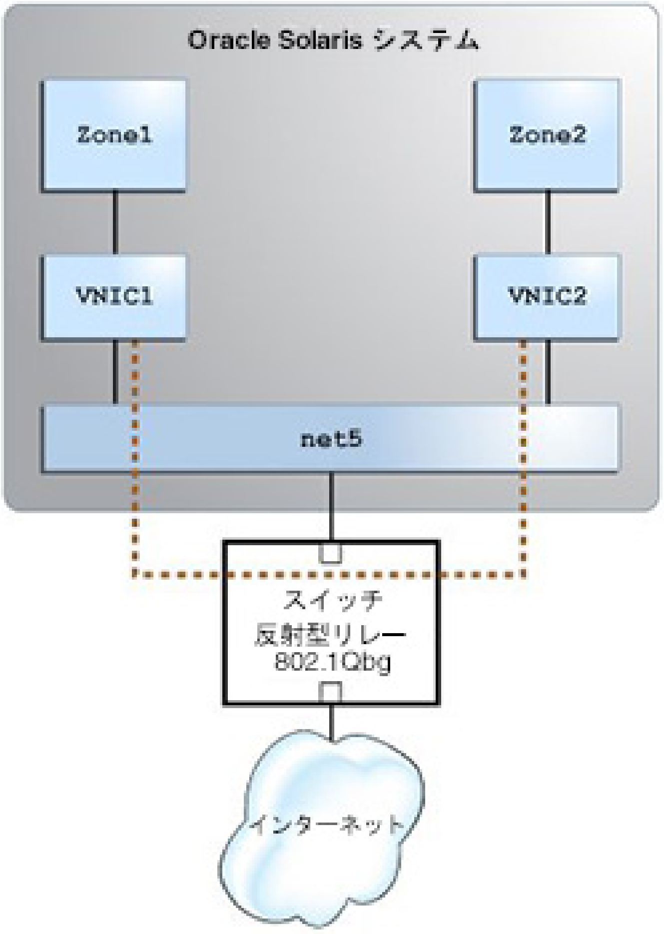
```
# dladm set-linkprop -p vswitchmode=remote net5

# dladm show-linkprop -p vswitchmode net5
LINK  PROPERTY     PERM  VALUE   EFFECTIVE  DEFAULT  POSSIBLE
net5  vswitchmode  rw    remote  remote     local    local,remote,auto
```

注記 - vswitchmode を remote に設定する前に、外部スイッチを反射型リレー用に構成する必要があります。

---

vswitchmode プロパティーを remote に設定して VNIC の内部通信を無効にするため、VNIC 間のネットワークトラフィックは、次の図に示すように外部スイッチ経由で送信されます。

図 4-4 外部スイッチを使用したゾーン間の通信



VM 間の通信を管理するための LLDP の使用

VM 間の通信の自動構成には LLDP を使用できます。LLDP は、外部スイッチが反射型リレーをサポートするかどうかに基づいて、ネットワークトラフィックの交換を内部または外部として構成します。LLDP を使用するには、`vswitchmode` データリンクプロパティーを `auto` に設定します。最初に、次の点を確認する必要があります。

- LLDP パッケージがインストールされている。

  LLDP パッケージがインストールされているかどうかを確認するには、次のコマンドを使用します。

  ```
  # pkg info lldp
  ```

- LLDP サービスがオンラインになっている。

  LLDP サービスがオンラインになっているかどうかを確認するには、次のコマンドを使用します。

  ```
  # svcs lldp
  STATE          STIME    FMRI
  online         Jul_13   svc:/network/lldp:default
  ```

- dot1-tlv TLV ユニットで EVB が有効になっている。
- NIC の LLDP モードが both である。

  この例では、EVB が dot1-tlv TLV ユニットで有効になっていて、LLDP モードが both であることを確認するため、次のコマンドを使用します。

```
# lldpadm show-agentprop -p mode,dot1-tlv net5
AGENT  PROPERTY  PERM  VALUE  DEFAULT  POSSIBLE
net5   mode      rw    both   disable  txonly,rxonly,both,disable
net5   dot1-tlv  rw    evb    none     none,vlanname,pvid,linkaggr,pfc,
                                       appln,evb,etscfg,etsreco,all
```

vswitchmode データリンクプロパティーを auto に設定するには:

```
# dladm set-linkprop -p vswitchmode=auto net5
```

vswitchmode データリンクプロパティーを auto に設定した場合は、dladm show-linkprop コマンドの出力を使用して、VM 間の通信が内部的か外部スイッチ経由かを確認できます。

```
# dladm show-linkprop -p vswitchmode net5
LINK  PROPERTY     PERM  VALUE   EFFECTIVE  DEFAULT  POSSIBLE
net5  vswitchmode  rw    auto    remote     local    local,remote,auto
```

この出力の EFFECTIVE フィールドの値が remote であるため、LLDP で外部スイッチ上の反射型リレーが有効になっており、VM 間の通信は外部スイッチ経由です。

LLDP についての詳細は、『Oracle Solaris 11.2 でのネットワークデータリンクの管理 』の第 5 章「リンク層検出プロトコルによるネットワーク接続情報の交換」を参照してください。

VDP を使用した VNIC 情報の交換

VNIC (VSI) 情報は、VDP (VSI Discovery and Configuration Protocol) を使用して、システム (ステーション) と外部スイッチ (ブリッジ) 間で交換されます。VDP Type-Length Value (TLV) ユニットは、ピア間の VDP パケットを確実に送信する、Edge Control Protocol (ECP) を使用して交換されます。VDP TLV ユニットは、VNIC の作成時または削除時に交換されます。

次の EVB コンポーネントは、システムが VNIC (VSI) 情報を外部スイッチに通知できるようにします。

- VSI プロファイルは、特定の VNIC 用に構成されているリンクプロパティーで構成されます。そのため、システムは構成済み VNIC と同じ数の VSI プロファイルを持つことができます。
- VSI 識別子は、VSI インスタンスを一意に識別します。Oracle Solaris では、この VSI インスタンスは VNIC (VSI) の MAC アドレスです。VSI タイプ ID と VSI バージョンは、特定の VSI マネージャー ID 内のプロファイルを識別します。
- VSI マネージャーは、VSI タイプ ID - VSI バージョンを特定の VNIC プロパティーセットとマッピングすることにより、システム上の複数の VSI プロファイルを管理します。Oracle Solaris では、デフォルトの VSI マネージャー `oracle_v1` が 3 バイトエンコードとして定義されています。この 3 バイトエンコードは、VDP パケットの Oracle Solaris ホストによって VSI タイプ ID として使用されます。
- VSI マネージャー ID は、特定の VSI タイプ ID - VSI バージョンペアに関連する VSI マネージャーを識別します。VSI マネージャー ID は IPv6 アドレスとして表されます。Oracle Solaris では、デフォルトの VSI マネージャー ID、`ORACLE_VSIMGR_V1` が定義されています。

注記 - 現時点で、VSI プロファイルとその特定のプロパティーを定義するための標準は定義されていません。VSI タイプの定義はベンダー固有であり、VSI マネージャー ID と密接にリンクされています。

この `oracle_v1` エンコーディングは、次のプロパティーをサポートします。

- 帯域幅の制限
- 帯域幅共有
- ベースとなるリンクのリンク速度
- VNIC の最大転送単位 (MTU)

Oracle Solaris では、システムは、oracle_v1 エンコーディングを使用してリンク情報をエンコードしてから情報を外部スイッチに送信します。スイッチが情報を受信すると、同じ oracle_v1 エンコーディングを使用して、エンコードされた情報をデコードします。

デフォルトでは、Oracle Solaris ホストは外部スイッチに次の要素を送信します。

- Oracle VSI マネージャー - oracle_v1
- VSI タイプ ID – oracle_v1 エンコーディングを使用してエンコードされた VNIC プロパティー
- VSI バージョン – 常に 0

Oracle Solaris では、VNIC 情報交換メカニズムは次のとおりです。

1. 外部スイッチは、Oracle VSI マネージャー oracle_v1 をサポートするように構成されています。
2. 外部スイッチは、oracle_v1 を使用して、VSI タイプ ID にエンコードされたプロパティーを確認します。
3. 外部スイッチは、その VNIC 用のプロパティー構成をパケットに適用します。

Oracle 組織固有の OUI TLV ユニットが VSI マネージャー ID TLV に続いて、それが Oracle 固有の VSI マネージャー ID であることを示します。スイッチからの応答に Oracle 固有の TLV ユニットが含まれていない場合は、Oracle Solaris ホストに対し、スイッチが Oracle VSI マネージャー (エンコーディング) をサポートしていないことを示します。Oracle Switch ES1-24 は、Oracle VSI マネージャー oracle_v1 をサポートします。Oracle Switch ES1-24 での EVB の構成についての詳細は、『*Sun Ethernet Fabric Operating System, EVB Administration Guide*』を参照してください。

注記 - VDP および ECP プロトコルのサポートに加えて、Oracle Solaris システムと相互運用するには、外部スイッチはデフォルトの Oracle VSI マネージャー ID である ORACLE_VSIMGR_V1 と、Oracle 組織固有の識別子 (OUI) TLV (エンコーディング情報を伝えるために使用されるサブタイプ VDP_ORACLEOUI_VSIMGR_SUBTYPE) もサポートする必要があります。

VDP による VNIC 情報の交換方法

VNIC 情報の交換は次のように行われます。

システムは、VNIC および関連するプロファイルを指定することで、外部スイッチに関連付けリクエスト (ASSOC) を送信します。外部スイッチは、成功または失敗の応答で関連付けリクエス

トに応答します。システムはそのあと、VNIC の関連付けを解除する関連付け解除リクエスト (DEASSOC) を外部スイッチに送信できます。VNIC のリクエストの状態を表示および取得する方法の詳細は、92 ページの「VDP と ECP の状態と統計情報」を参照してください。

VNIC を作成する際は、次のように VDP 交換が行われます。

1. VNIC に関する情報を含む VDP 関連付け (ASSOC) リクエストの TLV ユニットが、システムによって外部スイッチに送信されます。
2. 外部スイッチが VDP (ASSOC) TLV ユニットを受信し、VSI タイプ ID、VSI バージョン、および VSI マネージャー ID を使用して VNIC 情報を取得します。
3. 外部スイッチが、VNIC のプロパティー構成を適用します。
4. 外部スイッチは、VDP 関連付け (ASSOC) 応答の TLV ユニットをシステムに送信して、外部スイッチに VNIC のプロパティーが構成されたことを伝えます。

VNIC を削除する際は、次のように VDP 交換が行われます。

1. VSI ID を含む VDP 関連付けの解除 (DEASSOC) リクエストの TLV ユニットが、システムによって外部スイッチに送信されます。
2. 外部スイッチが VDP (DEASSOC) TLV ユニットを受信し、削除された VSI の VSI ID を取得します。
3. 外部スイッチが削除された VNIC の構成を削除します。
4. 外部スイッチが、VDP 関連付けの解除 (DEASSOC) 応答の TLV ユニットをシステムに送信します。

注記 - Oracle Solaris では、VDP は ASSOC および DEASSOC VDP リクエスト*のみ*をサポートします。

VDP と ECP の状態と統計情報

システムで EVB が有効になっている場合、さらに VNIC 用に VDP パケットが交換されている場合も、物理 Ethernet リンクの VDP 状態に関する情報を表示できます。単一のリンクに関する情報だけを表示するには、そのリンクをコマンドで指定します。それ以外の場合、すべての Ethernet リンクに関する VDP 情報が表示されます。

VDP の状態と統計情報の表示

VDP 状態を表示するには、次のコマンドを入力します。

```
# dladm show-ether -P vdp
VSI      LINK      VSIID              VSI-TYPEID   VSI-STATE   CMD-PENDING
vnic1    net0      2:8:20:22:3c:6b    98/0         ASSOC       NONE
vnic2    net0      2:8:20:90:7f:ef    96/0         ASSOC       NONE
```

VSI-STATE は、ピアとの VDP 交換のステータスを示します。指定できる値:

- TIMEDOUT – ピアは VDP リクエストに応答しませんでした。
- ASSOC – ピアはリクエストを正常に処理しました。
- DEASSOC – ホストまたはピアのいずれかがリクエストを拒否しました。ピアは、指定されたプロファイルまたはプロパティーを決定できない場合はリクエストを拒否できます。ホストは、oracle_v1 エンコーディングを使用していて、ピアの応答に Oracle OUI が含まれていない場合は、VDP パケットの交換を拒否できます。

サンプル出力は、リンク net0 上に 2 つの VSI (VNIC) が構成されていることを示しています。それぞれ固有の VSI ID は、それぞれに対応する MAC アドレスを示しています。VNIC の VSI-TYPE ID、vnic1 と vnic2 はそれぞれのプロパティー (帯域幅制限と MTU) から生成され、エンコーディングは oracle_v1 によって定義されます。

送信または受信 VDP パケットに関する統計情報を取得するには、次のコマンドを入力します。

```
# dlstat show-ether -P vdp net1
LINK     IPKTS    OPKTS    KeepAlives
net1     3        2        1
```

リンクプロパティーの表示

データリンクプロパティーを表示するには、dladm show-linkprop コマンドの -p オプションを使用します。

次の例は、vnic1 と vnic2 のリンクプロパティーを表示する方法を示しています。

```
# dladm show-linkprop -p maxbw,mtu vnic1
LINK      PROPERTY        PERM   VALUE     EFFECTIVE    DEFAULT   POSSIBLE
vnic1     maxbw           rw     100       100          --        --
vnic1     mtu             rw     1500      1500         1500      1500

# dladm show-linkprop -p maxbw,mtu vnic2
LINK      PROPERTY        PERM  VALUE      EFFECTIVE    DEFAULT   POSSIBLE
vnic2     maxbw           rw    20         20           --        --
vnic2     mtu             rw    1500       1500         1500      1500
```

ECP の状態と統計情報の表示

VDP は ECP を使用してメッセージを交換します。次の例は、物理リンク `net0` に固有の ECP の状態を示しています。

```
# dladm show-ether -P ecp net0
LINK        MAX-RETRIES     TIMEOUT
net0        3               164
```

`MAX-RETRIES`
: ピアから肯定応答を取得しないときに、ECP がパケットを送信する回数を指定します。

`TIMEOUT`
: パケットを再送信されるまでの間隔 (ミリ秒単位) を指定します。パケットを再送信するまで ECP が肯定応答を待機する時間間隔。

物理リンクの統計情報を取得するには、次のコマンドを入力します。

```
# dlstat show-ether -P ecp
LINK          IPKTS     OPKTS  IERRORS  OERRORS RETRANSMITS TIMEOUTS
net0          3         2      0        0       1           0
```

デフォルトの EVB 構成の変更

デフォルトでは、デフォルトの EVB 構成を変更する必要はありません。ほとんどの場合、EVB をインストールし、デフォルトの EVB 構成を使用すると、システムに外部スイッチを構成する VNIC に関する情報を交換できます。ただし、ホストとネットワークの EVB 構成を完全に制御して管理する場合は、デフォルトの構成を変更できます。

デフォルトの Oracle Solaris VSI マネージャー ID、`ORACLE_VSIMGR_V1` を使用すると、システムは自動的に、作成する VNIC の VSI タイプ ID を生成します。そのため、`vsi-typeid` や `vsi-vers` などのデータリンクプロパティーを設定する必要はありません。ただし、デフォルトの VSI マネージャー ID を使用しない場合は、`dladm set-linkprop` コマンドを使用して EVB に関連するデータリンクプロパティーを設定する必要があります。EVB に関連するデータリンクプロパティーを設定するには、外部スイッチがシステムと通信でき、VSI タイプ ID と VSI バージョンの特定のセットのプロパティーを取得できる必要があります。

EVB を使用するときにデフォルトの Oracle VSI マネージャー ID を使用して、Oracle VSI マネージャーがシステムの VSI プロファイルの VSI タイプ ID と VSI バージョンを自動的に生成できるようにします。

EVB に関連する次のデータリンクプロパティーを構成できます。

- `vsi-mgrid` は、物理リンクまたは VNIC に設定される VSI マネージャー ID を指定します。VNIC 用にこのプロパティーが設定されていない場合、基本となる物理リンクのデフォルト値 `ORACLE_VSIMGR_V1` が使用されます。

  `vsi-mgrid` プロパティーを明示的に設定する場合は、VSI タイプ ID と VSI バージョンも明示的に設定する必要があります。また、これらのプロパティーをデータリンク上に明示的に構成する必要もあります。

注記 - Oracle Solaris では、VSI マネージャー ID、VSI タイプ ID、および VSI バージョンを手動で構成する場合、対応する VNIC プロパティーは自動的には構成されません。

- `vsi-mgrid-enc` - VSI マネージャー ID に関連付けられているエンコーディングを示します。デフォルトでは、このプロパティーは `oracle_v1` に設定されます。VSI マネージャー ID に `oracle_v1` を関連付けない場合は、このプロパティー値を `none` に設定してください。値を `none` に設定する場合、VSI マネージャー ID、VSI タイプ ID、および VSI バージョンは自動的には生成されないため、それらは必ず手動で構成してください。
- `vsi-typeid` - VSI タイプ ID を指定します。VSI タイプ ID は、VSI バージョンとのペアで VSI プロファイルに関連付けられます。`vsi-mgrid` と `vsi-mgrid-enc` にデフォルト値を使用している場合、この 3 バイト値は自動的に生成されます。それ以外の場合は、このプロパティーの値を明示的に指定する必要があります。
- `vsi-vers` - VSI バージョンを指定します。VSI バージョンは、VSI タイプ ID とのペアで VSI プロファイルに関連付けられます。`vsi-mgrid` と `vsi-mgrid-enc` にデフォルト値を使用している場合、この 1 バイト値は自動的に生成されます。それ以外の場合は、このプロパティーの値を明示的に指定する必要があります。

EVB 関連のプロパティーは、`dladm show-linkprop` コマンドを使用して表示できます。VNIC 関連のリンクプロパティーの有効な値は、プロパティーのそれぞれの `EFFECTIVE` フィールドの値から取得できます。詳細は、例4-2「物理リンクでの EVB 関連のデータリンクプロパティーの表示例」 を参照してください。

EVB コンポーネントについての詳細は、90 ページの「VDP を使用した VNIC 情報の交換」を参照してください。EVB についての詳細は、evb(7P) のマニュアルページを参照してください。

▼ デフォルトの EVB 構成を変更する方法

`vsi-mgrid` および `vsi-mgrid-enc` プロパティーは、物理リンク上だけに構成する必要があります。`vsi-typeid` や `vsi-vers` などのその他の EVB 関連プロパティーは、VNIC 上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 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **プロファイルデータベースに記載されているデータリンクプロパティーを使用して、VNIC を作成します。**

   # dladm create-vnic -l *datalink* **-p maxbw=***maxbw-value***,priority=***priority-value VNIC*

3. **デフォルトの Oracle VSI マネージャー ID を使用しないため、物理リンクで VSI マネージャー ID に関連付けられているエンコーディングを none に設定します。**

   # dladm set-linkprop -p vsi-mgrid-enc=none *datalink*

4. **IPv6 アドレスを使用して物理リンク上の VSI マネージャー ID を設定します。**

   # dladm set-linkprop -p vsi-mgrid=*IPv6-address datalink*

5. **作成した VNIC の VSI タイプ ID と VSI バージョンを設定します。**

   # dladm set-linkprop -p vsi-typeid=*VSI-Type-ID***,vsi-vers=***VSI-Version VNIC*

6. **VNIC に設定されているプロパティーを確認します。**

   # dladm show-linkprop *VNIC*

例 4-1 EVB 関連のデータリンクプロパティーの設定

次の例は、EVB に関連するデータリンクプロパティーを設定する方法を示します。この例では、IPv6 アドレス、`IP1` を使用してアクセスできるプロファイルでシステムを使用しています。

VSI マネージャー ID、IP1 には、次のプロファイルが定義されています。

- VSI タイプ ID: 2
- VSI バージョン: 1

- データリンクプロパティー: maxbw=20、priority=5

1. プロファイルに記載されているデータリンクプロパティーを使用して、VNIC を作成します。

```
# dladm create-vnic -l net0 -p maxbw=20,priority=5 vnic1
```

2. デフォルトの Oracle VSI マネージャー ID を使用しないため、物理リンク net0 で VSI マネージャー ID に関連付けられているエンコーディングを none に設定します。

```
# dladm set-linkprop -p vsi-mgrid-enc=none net0
```

3. IPv6 アドレス IP1 を使用して、物理リンク net0 上の VSI マネージャー ID を設定します。

```
# dladm set-linkprop -p vsi-mgrid=IP1 net0
```

4. vnic1 の VSI タイプ ID と VSI バージョンを設定します。

```
# dladm set-linkprop -p vsi-typeid=2,vsi-vers=1 vnic1
```

5. vnic1 に設定されているプロパティーを確認します。

```
# dladm show-linkprop vnic1
LINK     PROPERTY            PERM VALUE      EFFECTIVE  DEFAULT    POSSIBLE
...
vnic1    vsi-typeid          rw   2          2           --         --
vnic1    vsi-vers            rw   1          1           --         --
vnic1    vsi-mgrid           rw   IP1        IP1         --         --
vnic1    vsi-mgrid-enc       rw   --         none        oracle_v1  none,oracle_v1
...
```

vnic1 の VDP ASSOC TLV ユニットには、次の情報が含まれます。

- VSI マネージャー ID = IP1
- VSI タイプ ID = 2
- VSI バージョン = 1

例 4-2 物理リンクでの EVB 関連のデータリンクプロパティーの表示例

次の例は、物理リンク上の EVB 関連のプロパティーを示しています。

```
# dladm show-linkprop -p vsi-mgrid,vsi-mgrid-enc net4
LINK     PROPERTY            PERM VALUE        EFFECTIVE    DEFAULT      POSSIBLE
net4     vsi-mgrid           rw   --           --           ::           --
net4     vsi-mgrid-enc       rw   --           --           oracle_v1    none,oracle_v1
```

この出力には、Oracle Solaris の EVB のデフォルト構成が表示されます。oracle_v1 エンコーディングを使用することで、VNIC 上に構成されているプロパティーから、VSI タイプ ID と VSI バージョンが自動的に生成されます。

例 4-3 VNIC での EVB 関連のプロパティーの表示

次の例は、VNIC 上の EVB 関連のプロパティーを示しています。

```
# dladm show-linkprop vnic0
LINK     PROPERTY              PERM VALUE      EFFECTIVE  DEFAULT    POSSIBLE
...
vnic0    vsi-typeid            rw   --         94         --         --
vnic0    vsi-vers              rw   --         0          --         --
vnic0    vsi-mgrid             rw   --         ::         --         --
vnic0    vsi-mgrid-enc         rw   --         oracle_v1  oracle_v1  none,oracle_v1
...
```

この出力には、vnic0 の有効なエンコーディングが oracle_v1 として表示されています。次に、vsi-typeid 94 の EFFECTIVE 値が自動的に生成され、vnic0 に対して有効になります。

♦ ♦ ♦ 第 5 章

エラスティック仮想スイッチについて

Oracle Solaris 11.2 リリース以降では、Oracle Solaris のエラスティック仮想スイッチ (Elastic Virtual Switch、EVS) 機能を使用して、複数の物理マシンにまたがる複数の仮想スイッチを管理できます。この章では、Oracle Solaris のエラスティック仮想スイッチ機能の概要を説明し、次のトピックが含まれます。

- 99 ページの「エラスティック仮想スイッチ (EVS) 機能の概要」
- 106 ページの「EVS のコンポーネント」
- 111 ページの「EVS 管理コマンド」
- 116 ページの「EVS を使用するための必須パッケージ」
- 117 ページの「ゾーンでの EVS の機能」
- 117 ページの「EVS を使用するためのセキュリティー要件」

エラスティック仮想スイッチ (EVS) 機能の概要

今日のデータセンターには、ネットワークファブリックで接続された複数の仮想マシン (VM) をホストする複数の物理サーバーがあります。データセンターへの VM 用のネットワークのプロビジョニングには、VM 間の仮想ネットワーキング、MAC アドレスと IP アドレスの管理、および VLAN と VXLAN の管理が含まれるため、管理者にとっては難題です。VM の内部および外部ネットワーク接続を確保すること以外の追加の課題として、VM および VM 内のアプリケーションのためのサービスレベル契約 (SLA) をプロビジョニングおよび適用することがあります。これらの SLA には、帯域幅制限や優先順位が含まれます。データセンター管理者は、共通のネットワークインフラストラクチャーを共有する複数のテナントを分離する必要もあります。

これらの要件を満たすために、Oracle Solaris ネットワーク仮想化により、管理者がデータセンター全体で仮想スイッチを管理できます。仮想スイッチは最上級のオペレーティングシステム抽象化として知られています。これらの仮想スイッチは、エラスティック仮想スイッチとも呼ばれ、複数の物理サーバーにまたがり、システム管理者が単一の仮想スイッチとして管理できます。

Oracle Solaris での仮想スイッチ

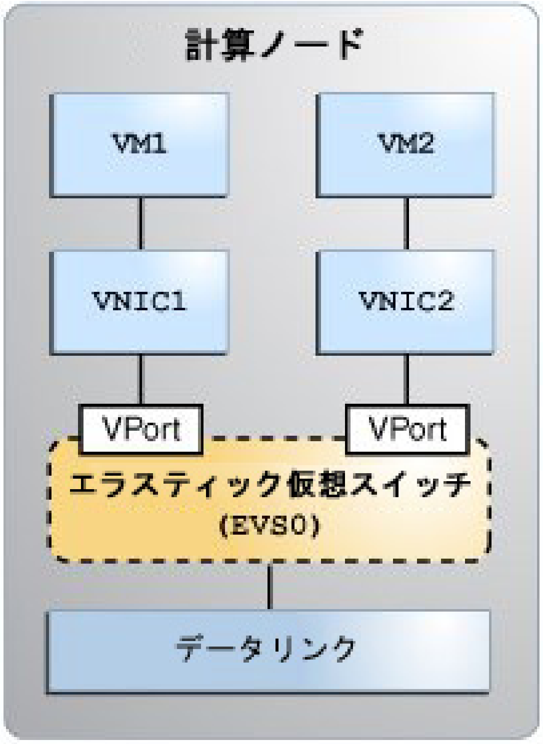
仮想スイッチは仮想マシン間の通信を容易にするエンティティーです。Oracle Solaris では、データリンク上に VNIC (リンクアグリゲーション、物理 NIC、etherstub など) を作成すると、仮想スイッチが自動的または暗黙的に作成されます。仮想スイッチは物理マシン内の VM 間のトラフィック (VM 間トラフィック) をループし、このトラフィックをネットワーク上では送信しません。すべての VM が相互に通信するには、同じレイヤー 2 セグメント上に存在する必要があります。詳細は、15 ページの「仮想スイッチ」を参照してください。

Oracle Solaris 11.2 より前のリリースでは、仮想スイッチは VNIC が作成されたデータリンク経由で間接的に管理されていました。Oracle Solaris 11.2 リリース以降、仮想スイッチは EVS で管理できます。仮想スイッチを明示的に作成して名前を指定したり、仮想スイッチに仮想ポート (VPort) を割り当てたり、IP アドレスのブロックに関連付けたりすることができます。仮想ポートに対しては、優先順位、最大帯域幅、サービスクラス (CoS)、MAC アドレス、IP アドレスなどのプロパティーを設定できます。また、仮想スイッチごとにデフォルトの SLA を構成することもできます。

注記 - VNIC 作成の一環として暗黙的に作成される仮想スイッチは、以前のリリースと同様にこのリリースでも引き続き存在し、機能します。EVS は既存の暗黙的な仮想スイッチを置き換えません。

次の図は、1 つの計算ノード内のエラスティック仮想スイッチ `EVS0` を示しています。

図 5-1 計算ノード内のエラスティック仮想スイッチ



Oracle Solaris のエラスティック仮想スイッチ機能とは

Oracle Solaris エラスティック仮想スイッチ (EVS) 機能により、1 つまたは複数の計算ノードにまたがる仮想スイッチを作成および管理できます。これらの計算ノードは VM をホストする物理マシンです。エラスティック仮想スイッチは、同じレイヤー 2 (L2) セグメントに属する明示的に作成された仮想スイッチを表すエンティティーです。エラスティック仮想スイッチは、ネットワーク上の任意の場所から接続された VM 間のネットワーク接続を提供します。

注記 - EVS では、仮想マシン (VM) という用語へのすべての参照が、具体的には Oracle Solaris ゾーンおよび Oracle Solaris カーネルゾーンを指します。

エラスティック仮想スイッチは、複数のホストにまたがることができます。これらの仮想スイッチには、ホストを組み込んだり、ホストを切り離したりする機能があるため、「エラスティック (柔軟性がある)」と呼ばれます。ホストの VNIC をエラスティック仮想スイッチに接続すると、エラスティック仮想スイッチはそのホストを組み込みます。これらの VNIC を削除すると、エラスティック仮想スイッチはそのホストを切り離します。

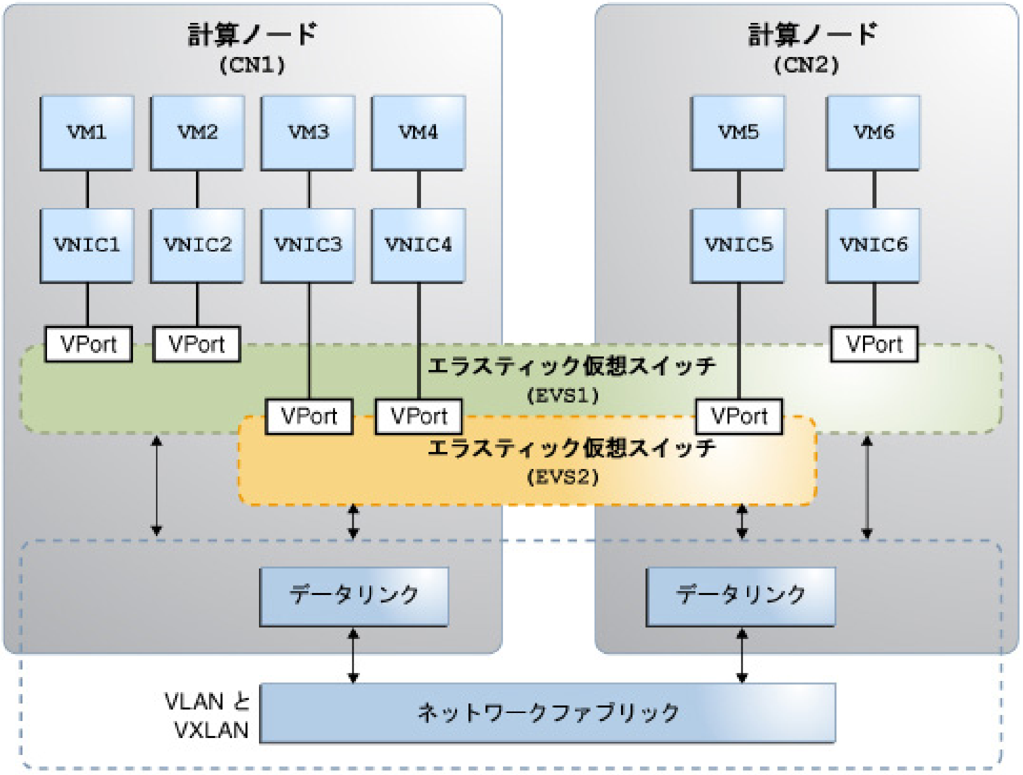
エラスティック仮想スイッチは分離された L2 セグメントを表し、その分離は VLAN または VXLAN を通して実装されます。VLAN でエラスティック仮想スイッチを実装する方法の詳細は、154 ページの「ユースケース: エラスティック仮想スイッチの構成」を参照してください。VXLAN でエラスティック仮想スイッチを実装する方法の詳細は、161 ページの「ユースケース: テナント用のエラスティック仮想スイッチの構成」を参照してください。

VLAN の管理については、『Oracle Solaris 11.2 でのネットワークデータリンクの管理 』の第 3 章「仮想ローカルエリアネットワークを使用した仮想ネットワークの構成」を参照してください。VXLAN の管理については、第3章「仮想拡張ローカルエリアネットワークを使用することによる仮想ネットワークの構成」を参照してください。

エラスティック仮想スイッチはすべて、名前、仮想ポート、および IP アドレスのブロックに関連付けられています。仮想スイッチのリソースを作成、モニター、および制御できます。詳細は、第6章「エラスティック仮想スイッチの管理」を参照してください。

次の図は、2 つの計算ノード間の 2 つのエラスティック仮想スイッチ (EVS1 および EVS2) を示しています。これらの計算ノードにプロビジョニングされる VM は、2 つの計算ノードにまたがるエラスティック仮想スイッチを介して接続されます。各計算ノードは、データリンク経由で同じネットワークファブリックに接続します。このデータリンクは、*アップリンクポート*とも呼ばれます。これらの計算ノード上のデータリンクは、仮想スイッチを外部ネットワークに接続します。VNIC は、仮想ポート (VPort) 経由でエラスティック仮想スイッチに接続されます。VNIC は、MAC アドレス、IP アドレス、SLA などの、仮想ポートに関連付けられたプロパティーを継承します。

図 5-2 計算ノード間のエラスティック仮想スイッチ



この図では、VM の VM1、VM2、および VM6 がエラスティック仮想スイッチ EVS1 経由で互いに通信できます。VM の VM3、VM4、および VM5 は、エラスティック仮想スイッチ EVS2 経由で互いに通信できます。詳細は、135 ページの「エラスティック仮想スイッチを構成する方法」を参照してください。

EVS を使用する利点

複数の仮想マシンをホストするデータセンター環境では、EVS は、次の利点を提供することでネットワーク管理タスクの一部をより単純にします。

- 複数のサーバー上にある VM 間の仮想ネットワークを作成して、ネットワーク接続する

- カスタムの SLA により仮想ポートの追加をサポートする
- VLAN または VXLAN を使用してネットワークを分離する
- ベースとなる同じインフラストラクチャーを共有するマルチテナント仮想ネットワークをサポートする
- Oracle Solaris ゾーンおよび Oracle Solaris カーネルゾーンと統合されている
- 次のものを一元管理する。
  - 仮想ポートの MAC アドレスと IP アドレス
  - 仮想スイッチごと、または仮想ポートごとの SLA
  - 仮想ポートの実行時ネットワークトラフィックの統計情報のモニタリング

エラスティック仮想スイッチのリソース

エラスティック仮想スイッチは、IP ネットワークと仮想ポートという主なリソースに関連付けられています。

IP ネットワーク

IP ネットワークは IPnet とも呼ばれ、IPv4 または IPv6 アドレスのブロックを、そのブロックのデフォルトルーターとともに表します。この IPv4 または IPv6 アドレスのブロックは、サブネットとも呼ばれます。エラスティック仮想スイッチには 1 つの IPnet しか関連付けることができません。仮想ポートを介してエラスティック仮想スイッチに接続するすべての VM は、エラスティック仮想スイッチに関連付けられている IPnet から IP アドレスが割り当てられます。

また、VPort の IP アドレスのプロパティー `ipaddr` を設定して、VM に IP アドレスを手動で割り当てることもできます。この IP アドレスは、IPnet のサブネット範囲内である必要があります。エラスティック仮想スイッチに IPnet を追加する方法の詳細は、135 ページの「エラスティック仮想スイッチを構成する方法」を参照してください。

仮想ポート

仮想ポートは VPort とも呼ばれ、VNIC とエラスティック仮想スイッチの間の接続点を表します。VNIC が VPort に接続すると、VNIC は VPort にカプセル化された次のようなネットワーク構成パラメータを継承します。

- 最大帯域幅、サービスクラス、優先度などの SLA パラメータ
- MAC アドレス
- IP アドレス

VPort を作成すると、ランダムに生成された MAC アドレスと、関連付けられた IPnet の次に使用可能な IP アドレスがその VPort に割り当てられます。ランダムに生成された MAC アドレスには、ローカルビットが設定された有効な IEEE OUI から成るデフォルトの接頭辞が付けられます。また、`evsadm add-vport` コマンドを使用して VPort を追加するときに、IP アドレスと MAC アドレスを指定することもできます。VPort を追加する方法の詳細は、135 ページの「エラスティック仮想スイッチを構成する方法」を参照してください。

注記 - エラスティック仮想スイッチに常に仮想ポートを追加する必要はありません。VNIC を作成するときに指定できるのは、その VNIC が接続する必要があるエラスティック仮想スイッチの名前だけです。このような場合は、EVS コントローラがシステムの仮想ポートを生成します。これらの仮想ポートは、`sys-`*vportname* という命名規則に従います (`sys-vport0` など)。システムの仮想ポートはエラスティック仮想スイッチのプロパティーを継承します。

次の表に、VPort のプロパティーを示します。

表 5-1 VPort のプロパティー

VPort のプロパティー	説明	使用可能な値	デフォルト値
`cos`	VPort 上のアウトバウンドパケットの 802.1p 優先度を指定します。	`0 - 7`	--
`maxbw`	VPort に全二重帯域幅を指定します。	--	--
`priority`	VPort に相対優先度を指定します。	`high`、`medium`、または `low`	`medium`
`ipaddr`	仮想ポートに関連付けられた IP アドレスを指定します。IP アドレスを割り当てることができるのは、VPort を作成するときだけです。	--	VPort の IP アドレスを指定しない場合は、EVS コントローラが自動的に、エラスティック仮想スイッチに関連付けられた IPnet から IP アドレスを選択します。
`macaddr`	VPort に関連付けられた MAC アドレスを指定します。MAC アドレスを割り当てることができるのは、	--	VPort の MAC アドレスを指定しない場合は、EVS コントローラが VPort のラ

VPort のプロパティー	説明	使用可能な値	デフォルト値
	VPort を作成するときだけです。		ンダムな MAC アドレスを生成します。
evs	VPort が関連付けられているエラスティック仮想スイッチを表す読み取り専用プロパティー。	--	--
tenant	VPort が関連付けられているテナントを表す読み取り専用プロパティー。	--	--

プロパティー evs と tenant は、読み取り専用プロパティーであるため変更できません。VPort のプロパティーについての詳細は、evsadm(1M) のマニュアルページを参照してください。

EVS での名前空間の管理

エラスティック仮想スイッチとそのリソースは論理的にグループ化されます。各論理グループは*テナント*と呼ばれます。テナント内のエラスティック仮想スイッチに対して定義されたリソースは、そのテナントの名前空間の外部では表示されません。テナントは、テナントのすべてのリソースをまとめて保持するためのコンテナとして機能します。テナントのエラスティック仮想スイッチを作成する方法の詳細は、135 ページの「エラスティック仮想スイッチを構成する方法」を参照してください。

EVS 操作にテナント名を指定する必要はありません。デフォルトのテナント名は sys-global で、すべての EVS 操作がこの名前空間で行われます。

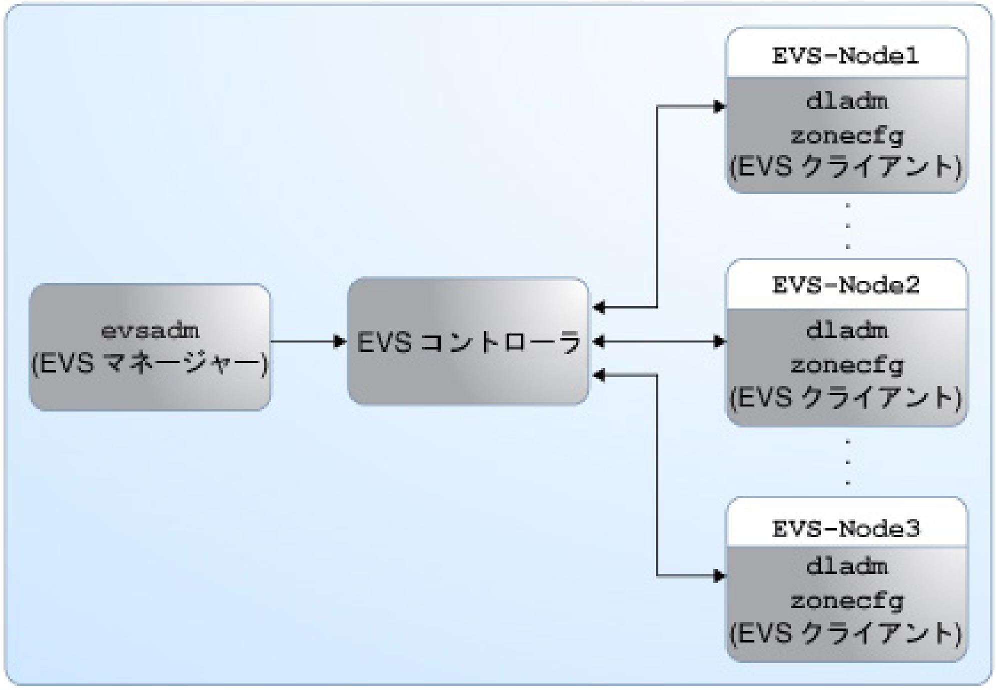
EVS のコンポーネント

EVS には次のコンポーネントがあります。

- EVS マネージャー
- EVS コントローラ
- EVS クライアント
- EVS ノード

次の図は、EVS のコンポーネントを示しています。

図 5-3 EVS のコンポーネント



この図では、EVS マネージャーと EVS コントローラは 2 つの個別のホストです。EVS ノード `EVS-Node1`、`EVS-Node2`、および `EVS-Node3` は、その VNIC またはゾーンの VNIC `anet` リソースがエラスティック仮想スイッチに接続する 3 つのホストです。

EVS マネージャー

EVS マネージャーは EVS コントローラと通信するエンティティーで、L2 ネットワークトポロジと、これらの L2 ネットワークで使用する必要のある IP アドレスを定義します。EVS マネージャーは、`evsadm` コマンドを使用して EVS コントローラと通信します。EVS マネージャーと EVS コントローラが同じ計算ノード上に存在することもできます。

注記 - L2 ネットワークトポロジはネットワークセグメントであり、各セグメントが、VLAN または VXLAN を使用して実装される 1 つのブロードキャストドメインを形成します。

`service/network/evs` パッケージをインストールし、`evsadm set-prop` コマンドで `controller` プロパティーを使用して EVS コントローラを指定したあと、EVS マネージャー上で EVS 操作を実行できます。`controller` プロパティーは、`ssh://[`*user*`@]example-controller.com` の形式で指定されます。詳細は、第6章「エラスティック仮想スイッチの管理」を参照してください。

EVS コントローラ

EVS コントローラは、エラスティック仮想スイッチとそれに関連付けられたすべてのリソースを構成および管理するための機能を提供します。データセンター内の EVS コントローラとして 1 つの物理マシンだけを設定する必要があります。

`evsadm set-prop` コマンドで `controller` プロパティーを使用して、EVS コントローラを指定します。`controller` プロパティーは `svc:/network/evs:default` SMF サービスに保存されるため、システムブートを通して保持されます。

EVS コントローラは、`evsadm set-controlprop` コマンドを使用して構成できるプロパティーに関連付けられます。複数の物理マシンにまたがる L2 セグメントを実装するには、使用可能な VLAN ID、使用可能な VXLAN セグメント ID、各 EVS ノードのアップリンクポートなどの情報を使用して EVS コントローラのプロパティーを構成する必要があります。EVS コントローラを構成し、そのプロパティーを設定する方法の詳細は、121 ページの「EVS コントローラの作成と管理」を参照してください。

注記 - SMF サイトプロファイルと自動インストール (AI) サービスを使用して、EVS コントローラの情報をデータセンター内の各 EVS ノードにプッシュすることもできます。SMF についての詳細は、『Oracle Solaris 11.2 でのシステムサービスの管理 』を参照してください。AI サービスの詳細は、『Oracle Solaris 11.2 システムのインストール 』の「インストールサービスの操作」を参照してください。

次の表は、EVS コントローラ用に構成できるプロパティーを示しています。

表 5-2 EVS コントローラのプロパティー

EVS コントローラのプロパティー	説明	使用可能な値	デフォルト値
`l2-type`	エラスティック仮想スイッチを物理マシンにまたがって実装する方法を定義します。	`vlan` または `vxlan`	`vlan`

EVS コントローラのプロパティー	説明	使用可能な値	デフォルト値
	注記 - l2-type プロパティーを変更しても、変更前に作成されたエラスティック仮想スイッチは影響を受けません。変更後に作成されたエラスティック仮想スイッチの l2-type プロパティーだけが更新されます。この動作は、EVS コントローラ内に VLAN および VXLAN に基づいた L2 セグメントが共存できることを意味します。		
vlan-range	エラスティック仮想スイッチの作成に使用する VLAN ID 範囲のコンマ区切りのリスト。各エラスティック仮想スイッチに 1 つの VLAN ID が関連付けられます。	1 - 4094	--
vxlan-range	エラスティック仮想スイッチの作成に使用する VXLAN セグメント番号の範囲のコンマ区切りのリスト。各エラスティック仮想スイッチに 1 つの VXLAN セグメント番号が関連付けられます。	0 - 16777215	--
vxlan-addr	VXLAN データリンクを作成する必要がある IP アドレスを指定します。サブネットに vxlan-addr プロパティーを設定することもできます。	--	--
vxlan-mgroup	VXLAN データリンクの作成中に使用する必要があるマルチキャストアドレスを指定します。	--	マルチキャストアドレスを指定しない場合、VXLAN データリンクは All Host アドレスを使用します。
vxlan-ipvers	VXLAN データリンクをホストする IP インタフェースに使用する必要があるアドレスの IP バージョンを指定します。	v4 または v6	v4
uplink-port	VLAN または VXLAN に使用する必要があるデータリンクを指定します。	--	--

EVS コントローラに設定するコントローラのプロパティーは、データセンター全体に適用されます。ただし、コントローラのプロパティー `uplink-port` および `vxlan-addr` の値はホストごとにオーバーライドできます。

たとえば、コントローラプロパティーを設定するときに、`uplink-port` プロパティーをデータセンター内のすべての EVS ノード上で VNIC または VXLAN の作成に使用されるデータリンク `net2` に設定したとします。ただし、データセンター内の EVS ノードにデータリンク `net1` が唯一のインタフェースとして存在する場合は、グローバル値 `net2` をホストごとの値で次のようにオーバーライドする必要があります。

```
# evsadm set-controlprop -h host1 -p uplink-port=net1
```

詳細は、130 ページの「EVS コントローラを構成する方法」を参照してください。

例6-2「EVS コントローラのプロパティーのリセット」 に示すように、コントローラプロパティーの値を指定しない場合、そのプロパティーはデフォルト値にリセットされます。EVS コントローラのプロパティーについての詳細は、`evsadm`(1M) のマニュアルページを参照してください。

EVS クライアント

`dladm` および `zonecfg` コマンドは EVS クライアントです。エラスティック仮想スイッチ、IPnet、および VPort を使用すると、`evsadm` コマンドによって L2 ネットワークトポロジを定義できます。`dladm` コマンドを使用して VNIC を L2 ネットワークトポロジに接続したり、または `zonecfg` コマンドを使用して VNIC `anet` リソースを接続し、それによってゾーンを L2 ネットワークトポロジに接続したりできます。

注記 - `evsadm` コマンドは L2 ネットワークトポロジを定義する EVS マネージャーです。

`dladm` コマンドまたは `zonecfg` コマンドを使用してエラスティック仮想スイッチの VNIC を作成すると、EVS コントローラから VNIC の構成情報が取得されます。

`service/network/evs` パッケージをインストールし、`evsadm set-prop` コマンドで `controller` プロパティーを使用して EVS コントローラを指定したあと、EVS クライアント上で EVS 操作を実行できます。`controller` プロパティーは、`ssh://[`*user*`@]example-controller.com` の形式で指定されます。詳細は、第6章「エラスティック仮想スイッチの管理」を参照してください。

EVS ノード

EVS ノードは、VNIC またはゾーンの VNIC anet リソースがエラスティック仮想スイッチに接続するホストです。dladm や zonecfg などのコマンドを使用すると、エラスティック仮想スイッチに接続する必要のある VNIC を指定できます。詳細は、137 ページの「エラスティック仮想スイッチ用の VNIC の作成」を参照してください。

EVS 管理コマンド

次の管理コマンドを使用して、エラスティック仮想スイッチを管理します。

- evsadm
- evsstat
- dladm
- zonecfg

エラスティック仮想スイッチを構成する方法の詳細は、135 ページの「エラスティック仮想スイッチを構成する方法」を参照してください。

evsadm コマンド

EVS コントローラと通信したり、エラスティック仮想スイッチ、IPnet、および VPort を管理するには、evsadm コマンドを使用します。このセクションでは、このコマンドで操作を実行するために使用するサブコマンドについて説明します。詳細は、evsadm(1M) のマニュアルページを参照してください。

エラスティック仮想スイッチを管理するための evsadm サブコマンド

仮想スイッチを管理するための evsadm サブコマンドは次のとおりです。

create-evs	エラスティック仮想スイッチを作成します
delete-evs	エラスティック仮想スイッチを削除します

show-evs	エラスティック仮想スイッチに関する情報を表示します
set-evsprop	エラスティックスイッチの maxbw および priority プロパティーを設定できるようにします これらのプロパティーについての詳細は、141 ページの「エラスティック仮想スイッチのプロパティーの設定」を参照してください。
show-evsprop	エラスティック仮想スイッチのプロパティーを表示します

IPnet を管理するための evsadm サブコマンド

IPnet を管理するための evsadm サブコマンドは次のとおりです。

add-ipnet	エラスティック仮想スイッチに IPnet を追加し、subnet および defrouter プロパティーを設定できるようにします これらのプロパティーの詳細は、134 ページの「エラスティック仮想スイッチへの IPnet の追加」を参照してください。
remove-ipnet	IPnet を削除します
show-ipnet	IPnet に関する情報を表示します

VPort を管理するための evsadm サブコマンド

仮想ポートを管理するための evsadm サブコマンドは次のとおりです。

add-vport	VPort を追加します
remove-vport	VPort を削除します
show-vport	VPort に関する情報を表示します
set-evsprop	VPort の次のプロパティーを設定できるようにします ■ cos ■ maxbw ■ priority これらのプロパティーについての詳細は、表5-1「VPort のプロパティー」を参照してください。
show-vportprop	VPort のプロパティーを表示します

reset-vport	VPort をリセットします

EVS クライアントプロパティーを管理するための evsadm サブコマンド

EVS クライアントプロパティーを管理するための evsadm サブコマンドは次のとおりです。

set-prop	controller プロパティーを設定できるようにします
show-prop	EVS クライアントプロパティーを表示します

EVS コントローラプロパティーを管理するための evsadm サブコマンド

EVS コントローラプロパティーを管理するための evsadm サブコマンドは次のとおりです。

set-controlprop

コントローラの次のプロパティーを設定できるようにします

- l2-type
- vlan-range
- vxlan-range
- vxlan-mgroup
- vxlan-addr
- vxlan-ipvers
- uplink-port

これらのプロパティーについての詳細は、表5-2「EVS コントローラのプロパティー」 を参照してください。

show-controlprop

EVS コントローラのプロパティーを表示します

evsstat コマンド

evsstat コマンドは、データセンター内のすべての VPort、または指定されたエラスティック仮想スイッチのすべての VPort の、ネットワークトラフィック統計情報を表示します。また、VPort に関連付けられた VNIC の統計情報も報告します。詳細は、151 ページの「エラスティック仮想スイッチのモニタリング」を参照してください。evsstat コマンドについての詳細は、evsstat(1M) のマニュアルページを参照してください。

dladm コマンド

エラスティック仮想スイッチに接続された VNIC は、次の dladm コマンドを使用して管理できます。

- dladm create-vnic コマンド - VNIC を作成し、VNIC を接続する必要があるエラスティック仮想スイッチ名を指定できます。オプションで、エラスティック仮想スイッチの VPort を指定できます。
- dladm show-vnic コマンド – 特定の VNIC のエラスティック仮想スイッチ情報を表示できます。dladm show-vnic コマンドの出力には、フィールド TENANT、EVS、および VPORT も表示されます。ただし、これらのフィールドはゾーン内では不可視になります。

詳細は、dladm(1M) のマニュアルページを参照してください。

エラスティック仮想スイッチ用の VNIC を構成する方法の詳細は、137 ページの「エラスティック仮想スイッチの VNIC を作成する方法」を参照してください。

zonecfg コマンド

拡張された zonecfg コマンドを使用して、エラスティック仮想スイッチのゾーンの VNIC anet リソースを構成します。VNIC anet リソースに次のプロパティーを設定できます。

- tenant – テナントの名前を指定します。ゾーンを構成するときに値を指定しないと、システムがデフォルト値、sys-global を割り当てます。
- vport – VPort の名前を指定します。ゾーンを構成するときに値を指定しないと、システムがエラスティック仮想スイッチの VPort を生成し、VPort はエラスティック仮想スイッチのプロパティーを継承します。
- evs – VNIC anet リソースを接続する必要のあるエラスティック仮想スイッチの名前を指定します。

anet リソースについての詳細は、『Oracle Solaris ゾーンの紹介 』の「リソースタイプのプロパティー」の anet の説明を参照してください。

注記 - ゾーン構成には、テナント名、エラスティック仮想スイッチ名、およびデータセンター内の VPort が一意に識別される VPort 名が含まれる必要があります。ゾーン構成についての詳細は、『Oracle Solaris ゾーンの作成と使用 』を参照してください。

エラスティック仮想スイッチ用に VNIC anet リソースを構成する方法の詳細は、138 ページの「エラスティック仮想スイッチの VNIC anet リソースの作成」を参照してください。zonecfg コマンドについての詳細は、zonecfg(1M) のマニュアルページを参照してください。

エラスティック仮想スイッチに接続された VNIC を管理する場合の制限

dladm create-vnic コマンドまたは zonecfg コマンドを使用してエラスティック仮想スイッチを作成して接続する VNIC には、次の制限が適用されます。

- VNIC の名前の変更に、dladm rename-link コマンドを使用することはできません。
- そのような VNIC のプロパティーの変更に、dladm set-linkprop または dladm reset-linkprop コマンドを使用することはできません。
- これらの VNIC の変更に、dladm modify-vnic コマンドを使用することはできません。

自動的に生成される VXLAN データリンク

VXLAN を使用してエラスティック仮想スイッチのレイヤー 2 セグメントを実装した場合は、エラスティック仮想スイッチの VNIC をホストする EVS ノード上に VXLAN データリンクが EVS によって自動的に作成されます。これらのデータリンクは、自動的に生成される VXLAN データリンクと呼ばれ、evs-vxlan*segment-ID* という命名規則に従います。ここで、evs はそのデータリンクを作成したエンティティーです。たとえば、evs-vxlan200 という名前は、200 が VXLAN ID であり、evs がこのデータリンクを作成したエンティティーであることを示します。dladm show-vxlan コマンドを使用すると、自動的に生成される VXLAN データリンクを表示できます。詳細は、72 ページの「VXLAN 情報の表示」を参照してください。

自動的に生成される VXLAN データリンクに対して dladm サブコマンドを使用して、そのデータリンクを削除したり、名前を変更したりすることはできません。ただし、dladm set-linkprop コマンドと dladm reset-linkprop コマンドを使用して、データリンクプロパティーを一時的に設定できます。

EVS を使用するための必須パッケージ

EVS を使用する前に次のパッケージをインストールする必要があります。

- `pkg:/service/network/evs`

  EVS マネージャー、EVS コントローラ、および EVS ノードにコアパッケージ `pkg:/service/network/evs` をインストールする必要があります。このパッケージには次のコンポーネントが含まれます。

  - `evsadm`
  - `evsstat`
  - SMF サービス (`svc:/network/evs:default`) – この SMF サービスには、EVS コントローラのホスト名または IP アドレスを保持する `controller` プロパティーがあります。EVS クライアントは、ホスト名または IP アドレスを使用して EVS コントローラと通信します。`controller` プロパティーを管理するには、`evsadm set-prop` コマンドを使用します。

  `pkg:/service/network/evs` パッケージをインストールすると、新しいユーザー `evsuser` が作成されます。`evsuser` は、Elastic Virtual Switch Administration 権利プロファイルを持つ特定のユーザーです。このプロファイルは、EVS 操作を実行するために必要なすべての承認と特権を提供します。

- `pkg:/system/management/rad/module/rad-evs-controller`

  このパッケージは、EVS コントローラとして機能するシステムだけにインストールする必要があります。データセンター内のすべてのエラスティック仮想スイッチを管理するには、1 つのコントローラのみを使用する必要があります。このパッケージには SMF サービス、`svc:/network/evs-controller:default` が含まれます。この SMF サービスには、物理マシンにまたがる L2 セグメントを実装するために必要な情報を取得するプロパティーがあります。コントローラのプロパティーを管理するには、`evsadm set-controlprop` コマンドを使用します。

詳細は、122 ページの「EVS コントローラの必須パッケージ」を参照してください。

ゾーンでの EVS の機能

`zonecfg` コマンドに関連付けられたプロパティーを使用すると、エラスティック仮想スイッチに VNIC `anet` リソースを接続できます。Oracle Solaris ゾーンと Oracle Solaris カーネルゾーンは EVS 機能をサポートします。

カーネルゾーンは、エラスティック仮想スイッチ用に作成する VNIC をサポートします。カーネルゾーン内に作成する VNIC は、VNIC が `zvnet` ドライバに関連付けられたファクトリ MAC アドレスを使用している場合にのみ機能します。エラスティック仮想スイッチ用に作成する VNIC はそのエラスティック仮想スイッチの VPort に関連付けられた MAC アドレスを継承するため、エラスティック仮想スイッチの VPort は、`macaddr` プロパティーを `zvnet` ドライバのファクトリ MAC アドレスに設定することによって作成する必要があります。

ファクトリ MAC アドレスを明示的に指定するには、次のコマンド構文を使用します。

```
# evsadm add-vport -p macaddr=factory-MAC-addr-zvnet EVS-name/VPort-name
```

カーネルゾーンでは、このコマンドを使用して作成された VPort に VNIC を接続できます。カーネルゾーンについては、『Oracle Solaris カーネルゾーンの作成と使用 』を参照してください。

EVS を使用するためのセキュリティー要件

EVS 操作を実行するには、スーパーユーザーか、または Elastic Virtual Switch Administration 権利プロファイルを持つユーザーである必要があります。また、ユーザーを作成し、そのユーザーに Elastic Virtual Switch Administration 権利プロファイルを割り当てることもできます。詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』を参照してください。

注記 - マルチテナント EVS 設定では、各ユーザーに対するテナントごとのユーザー承認がサポートされていないため、個々のテナントが独自のエラスティック仮想スイッチやそのリソースを管理することはできません。EVS ドメイン全体で、すべてのテナントのリソースを管理する管理者は 1 人である必要があります。

次の例は、Elastic Virtual Switch Administration 権利プロファイルを持つ `user1` を作成する方法を示しています。

```
# useradd -P “Elastic Virtual Switch Administration” user1
```

次の例は、既存のユーザー `user1` に Elastic Virtual Switch Administration 権利プロファイルを追加する方法を示しています。

```
# usermod -P +"Elastic Virtual Switch Administration" user1
```

EVS コントローラを設定する場合は、Elastic Virtual Switch Administration 権利プロファイルを持つユーザーを指定する必要があります。たとえば、EVS コントローラを設定する場合は、`user1` を次のように指定する必要があります。

```
# evsadm set-prop -p controller=ssh://user1@example-controller.com
```

詳細は、125 ページの「EVS コントローラの構成」を参照してください。

注記 - また、`pkg:/service/network/evs` パッケージをインストールしたときに作成される `evsuser` を使用することもできます。ユーザー `evsuser` には、Elastic Virtual Switch Administration 権利プロファイルが割り当てられます。このプロファイルは、EVS 操作を実行するために必要なすべての承認と特権を提供します。

♦ ♦ ♦ 第 6 章

エラスティック仮想スイッチの管理

この章では、エラスティック仮想スイッチとそのリソースの管理タスクについて説明します。一般的な情報については、第5章「エラスティック仮想スイッチについて」を参照してください。

この章の内容は、次のとおりです。

- 119 ページの「EVS 管理タスク」
- 120 ページの「エラスティック仮想スイッチ構成の計画」
- 121 ページの「EVS コントローラの作成と管理」
- 132 ページの「エラスティック仮想スイッチの構成」
- 139 ページの「エラスティック仮想スイッチ、IPnet、および VPort の管理」
- 151 ページの「エラスティック仮想スイッチのモニタリング」
- 154 ページの「エラスティック仮想スイッチのユースケースの例」

EVS 管理タスク

このセクションでは、EVS 管理タスクを実行するための次の情報について説明します。

- 130 ページの「EVS コントローラを構成する方法」
- 135 ページの「エラスティック仮想スイッチを構成する方法」
- 137 ページの「エラスティック仮想スイッチ用の VNIC の作成」
- 140 ページの「エラスティック仮想スイッチ情報の表示」
- 141 ページの「エラスティック仮想スイッチのプロパティーの設定」
- 142 ページの「エラスティック仮想スイッチのプロパティーの表示」
- 144 ページの「IPnet の削除」
- 144 ページの「IPnet の表示」
- 146 ページの「VPort のプロパティーの設定」

- 146 ページの「VPort のプロパティーの表示」
- 148 ページの「VPort の表示」
- 150 ページの「VPort の削除」
- 150 ページの「エラスティック仮想スイッチを削除する方法」
- 151 ページの「エラスティック仮想スイッチのモニタリング」

エラスティック仮想スイッチ構成の計画

エラスティック仮想スイッチ構成の計画には次のアクションが含まれます。

1. EVS コントローラ、EVS マネージャー、および EVS ノードへの必須パッケージのインストール。これらのパッケージは、これらの各コンポーネントに個別にインストールする必要があります。詳細は、116 ページの「EVS を使用するための必須パッケージ」を参照してください。
2. EVS 設定の次のコンポーネント間での `evsuser` の事前共有された公開鍵を使用した SSH 認証の設定。
   - EVS マネージャーと EVS コントローラ
   - 各 EVS ノードと EVS コントローラ
   - EVS コントローラと各 EVS ノード

   詳細は、126 ページの「SSH 認証の設定」を参照してください。
3. `controller` プロパティーを設定することによって EVS コントローラを指定します。EVS ノード、EVS マネージャー、および EVS コントローラに、EVS コントローラのホスト名または IP アドレスを指定する必要があります。詳細は、125 ページの「EVS コントローラの構成」を参照してください。
4. 次のような、EVS コントローラの構成。

   a. EVS コントローラのプロパティーの設定。
   b. EVS コントローラ用に設定されているプロパティーの確認。

   詳細は、130 ページの「EVS コントローラを構成する方法」を参照してください。
5. EVS マネージャーを使用したエラスティック仮想スイッチの構成。これには、次の操作が含まれます。

   a. エラスティック仮想スイッチの作成。
   b. エラスティック仮想スイッチへの IPnet の追加。

c. エラスティック仮想スイッチへの VPort の追加。

d. 構成したエラスティック仮想スイッチの確認。

詳細は、135 ページの「エラスティック仮想スイッチを構成する方法」を参照してください。

6. 次のような、EVS ノードでの VNIC の作成およびエラスティック仮想スイッチへの VNIC の接続。

a. `dladm` コマンドの使用による VNIC の作成、または `zonecfg` コマンドの使用による VNIC `anet` リソースの作成とエラスティック仮想スイッチへの接続。

b. エラスティック仮想スイッチに接続された VNIC の確認。

詳細は、137 ページの「エラスティック仮想スイッチ用の VNIC の作成」を参照してください。

EVS コントローラの作成と管理

EVS コントローラは、エラスティック仮想スイッチとそれに関連付けられたすべてのリソースの構成および管理のための機能を提供します。物理マシンにレイヤー 2 セグメントを実装するために必要な情報を取得する、EVS コントローラのプロパティーを設定する必要があります。詳細は、108 ページの「EVS コントローラ」を参照してください。

EVS コントローラの計画には次の考慮事項があります。

- VLAN、VXLAN、またはその両方のいずれを使用してエラスティック仮想スイッチを実装するかを決定します。
  - VLAN を使用してエラスティック仮想スイッチを実装する場合は、プロパティー `uplink-port` および `vlan-range` を設定する必要があります。
  - VXLAN を使用してエラスティック仮想スイッチを実装する場合は、プロパティー `vxlan-range` および `uplink-port` または `vxlan-addr` を設定する必要があります。オプションで、プロパティー `vxlan-mgroup` および `vxlan-ipvers` を設定することもできます。

注記 - エラスティック仮想スイッチを作成したあとに、そのエラスティック仮想スイッチの EVS コントローラプロパティーを変更することはできません。EVS コントローラプロパティーへのすべての変更が、今後作成する新しいエラスティック仮想スイッチに反映されます。

- 計算ノードに同じデータリンクがない場合は、計算ノードごとに uplink-port プロパティーに対してデータリンクを指定する必要があります。

  たとえば、データリンク net2 を持つ host1 と、データリンク net3 を持つ host2 の 2 つの計算ノードがあるとします。uplink-port プロパティーを設定する場合は、両方のホストのデータリンクを次のように指定する必要があります。

  ```
  # evsadm set-controlprop -h host1 -p uplink-port=net2
  # evsadm set-controlprop -h host2 -p uplink-port=net3
  ```

EVS コントローラの必須パッケージ

データセンター内のすべてのエラスティック仮想スイッチを管理するには、1 つのコントローラのみを使用する必要があります。EVS コントローラとして機能するシステムに、pkg:/service/network/evs パッケージと pkg:/system/management/rad/module/rad-evs-controller パッケージをインストールする必要があります。

パッケージをインストールするには、次のコマンドを使用します。

```
# pkg install evs
# pkg install rad-evs-controller
```

rad-evs-controller パッケージをインストールしたあと、次のコマンドを使用して rad:local サービスを再起動して、EVS コントローラをロードする必要があります。

```
# svcadm restart rad:local
```

EVS コントローラを構成するためのコマンド

このセクションでは、EVS コントローラ用の次のタスクを実行する方法について説明します。

- EVS コントローラの設定
- EVS コントローラの表示
- EVS コントローラのプロパティーの設定
- EVS コントローラのプロパティーの表示

EVS コントローラの設定

ホスト上の EVS コントローラを設定するには、evsadm set-prop コマンドを使用します。コマンドの構文は次のとおりです。

```
# evsadm set-prop -p controller=[value[...,]]
```

このコマンドは、コマンドが実行されるホストのプロパティーの値を設定します。サポートされている唯一のプロパティーは controller であり、ssh://[*user*@]evs-controller-*host-name* または ssh://[*user*@]evs-controller-*IP-address* の形式で指定できます。

EVS コントローラの表示

EVS コントローラを表示するには、evsadm show-prop コマンドを使用します。コマンドの構文は次のとおりです。

```
# evsadm show-prop [[-c] -o field[,...]] [-p controller[,...]]
```

-p controller
RAD クライアントが接続する必要がある EVS コントローラを指定します。

-o *field*[,...]
表示する出力フィールドのコンマ区切りのリストを指定します (大文字と小文字は区別されません)。出力に列として表示される次のフィールドを指定できます。

all	すべての出力フィールドを表示します
PROPERTY	プロパティーの名前
PERM	プロパティーの権限 (rw または r- のいずれか)
VALUE	プロパティーの値
DEFAULT	プロパティーのデフォルト値

-c
安定したマシン解析可能な形式を使用して表示します。-c オプションとともに -o オプションを指定する必要があります。

EVS コントローラを表示する方法を示す例については、例6-1「EVS コントローラの構成」を参照してください。

EVS コントローラのプロパティーの設定

EVS コントローラのプロパティーを設定するには、`evsadm set-controlprop` コマンドを使用します。コマンドの構文は次のとおりです。

```
# evsadm set-controlprop [-h host] -p prop=[value[...,]]
```

`-h` *host* — プロパティーが設定されるホストを指定します。

`-p` *prop* — EVS コントローラに設定されるコントローラのプロパティーの名前を指定します。プロパティーが複数の値を取る場合は、区切り文字としてコンマを使用して値を指定する必要があります。一度に 1 つのプロパティーだけを指定する必要があります。値が指定されない場合、プロパティーはデフォルト値にリセットされます。EVS コントローラに設定できるプロパティーについての詳細は、表5-2「EVS コントローラのプロパティー」を参照してください。

EVS コントローラのプロパティーの表示

EVS コントローラのプロパティーを表示するには、`evsadm show-controlprop` コマンドを使用します。コマンドの構文は次のとおりです。

```
# evsadm show-controlprop [[-c] -o field[,...]] [-p prop[,...]]
```

このコマンドは、EVS コントローラの 1 つ以上のプロパティーの現在の値を表示します。EVS コントローラのプロパティーが指定されていない場合は、コントローラの既存のすべてのプロパティーが表示されます。コントローラのプロパティーについての詳細は、表5-2「EVS コントローラのプロパティー」を参照してください。

`-o` *field*[,...] — 表示する出力フィールドのコンマ区切りのリストを指定します (大文字と小文字は区別されません)。出力に列として表示される次のフィールドを指定できます。

- `all` — すべての出力フィールドを表示します。
- `PROPERTY` — プロパティーの名前。
- `PERM` — プロパティーの権限 (`rw` または `r-` のいずれか)。
- `VALUE` — プロパティーの値。
- `DEFAULT` — プロパティーのデフォルト値。

HOST	値が -- の場合、プロパティーはグローバルであり、すべてのホストに適用されます。それ以外の場合、プロパティーは特定のホストに適用されます。

EVS コントローラのプロパティーを表示する方法を示す例については、例6-1「EVS コントローラの構成」 を参照してください。

EVS コントローラの構成

ネットワークで EVS コントローラとして計算ノードを 1 つだけ構成してから、EVS ノードが EVS コントローラと通信できるように各 EVS ノード上に EVS コントローラを設定する必要があります。ただし、EVS コントローラのプロパティーは、その EVS コントローラと通信できるいずれかのノードから 1 回設定するだけで済みます。EVS コントローラのプロパティーを設定するには、`evsadm set-controlprop` コマンドを使用します。詳細は、130 ページの「EVS コントローラを構成する方法」を参照してください。

EVS コントローラのプロパティーをリセットすることもできます。例6-2「EVS コントローラのプロパティーのリセット」 は、EVS コントローラのプロパティーをリセットする方法を示しています。EVS コントローラとそのプロパティーの詳細は、108 ページの「EVS コントローラ」を参照してください。

エラスティック仮想スイッチの構成を簡素化するには、`evsuser` として接続する必要があります。必須の EVS パッケージ (`service/network/evs`) をインストールすると、特殊なユーザーである `evsuser` が作成され、Elastic Virtual Switch Administration 権利プロファイルが割り当てられます。このプロファイルには、EVS 操作を実行するためのすべての承認と特権が含まれています。`evsuser` を使用するには、`controller` プロパティーを次のように設定する必要があります。

```
# evsadm set-prop -p controller=ssh://evsuser@evs-controller-hostname-or-IP-address
```

さらに、`evsadm` コマンドを実行するホストと EVS コントローラの間に事前共有された公開鍵を使用した SSH 認証を設定する必要があります。

注記 - EVS 操作を実行するには、スーパーユーザーか、または Elastic Virtual Switch Administration 権利プロファイルを持つユーザーである必要があります。詳細は、117 ページの「EVS を使用するためのセキュリティー要件」を参照してください。

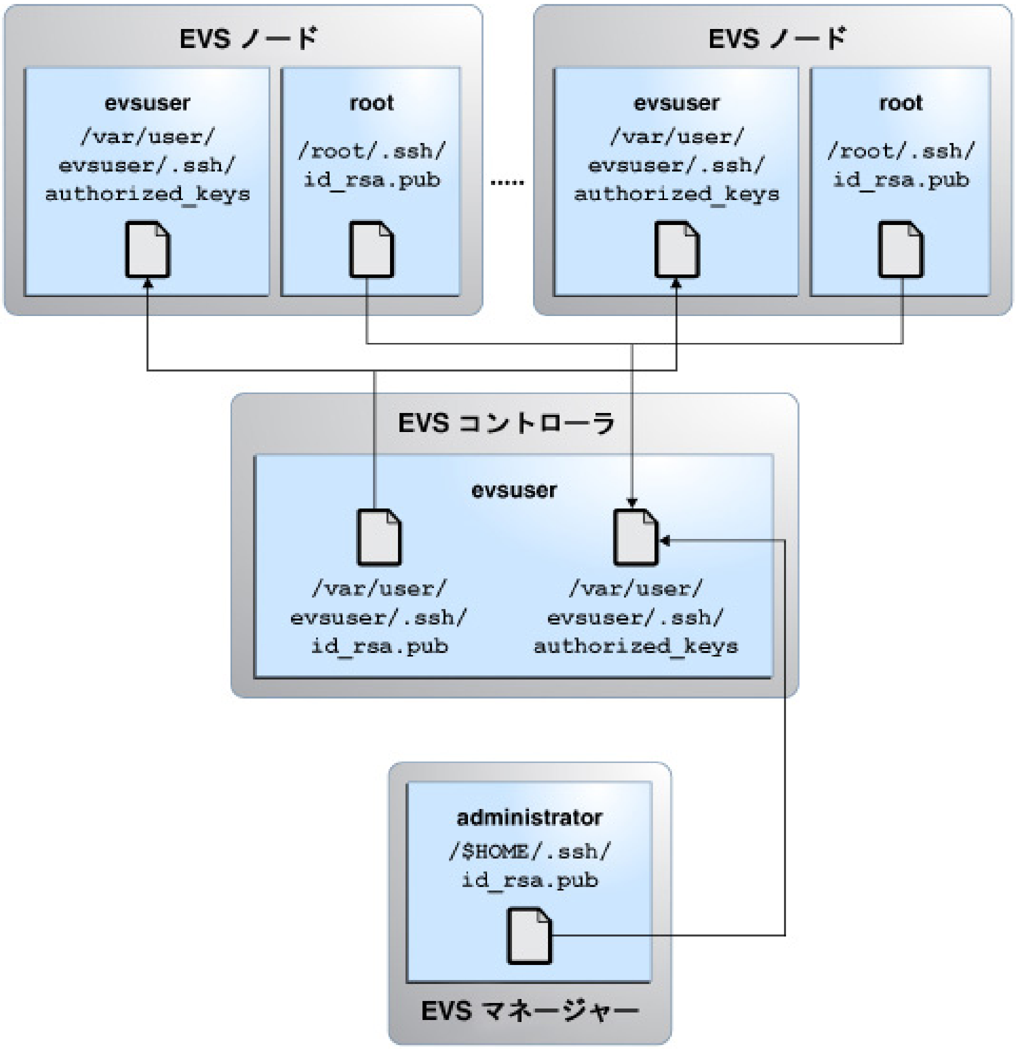
SSH 認証の設定

evsadm コマンドが EVS コントローラと非対話形式で、セキュアに通信できるようにするには、事前共有された公開鍵を使用した SSH 認証が必要です。EVS 設定の次のコンポーネント間に evsuser の事前共有された公開鍵を使用した SSH 認証を設定する必要があります。

- **EVS マネージャーと EVS コントローラ** – EVS マネージャー上で evsadm コマンドを実行している管理者またはユーザーの公開鍵を EVS コントローラ上の /var/user/evsuser/.ssh/authorized_keys ファイル内に追加します。
- **EVS ノードと EVS コントローラ** – 各 EVS ノード上の root ユーザーの公開鍵を EVS コントローラ上の /var/user/evsuser/.ssh/authorized_keys ファイル内に追加します。これらの公開鍵を追加する必要があるのは、zoneadmd デーモンが root として動作するためです。このデーモンは EVS コントローラに接続し、VNIC anet リソースの構成情報を取得します。詳細は、zoneadmd(1M) のマニュアルページを参照してください。
- **EVS コントローラと EVS ノード** – EVS コントローラは VPort プロパティーを設定するために各 EVS ノードと通信するため、EVS コントローラ上の evsuser の公開鍵を各 EVS ノード上の /var/user/evsuser/.ssh/authorized_keys ファイル内に追加します。

次の図は、EVS コンポーネント間の SSH 認証の設定を示しています。

図 6-1 EVS 設定での SSH 認証



SSH 認証を設定したあと、EVS コントローラを指定する必要があります。ここでは、EVS ノード、EVS マネージャー、および EVS コントローラ上で `controller` プロパティーが `ssh://evsuser@evs-controller.example.com` に設定されていることを前提にしています。

次の手順は、SSH 認証を設定する方法を示しています。

▼ EVS ノードと EVS コントローラの間に SSH 認証を設定する方法

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **EVS ノードで RSA 鍵ペアを生成します。**

```
evs-node# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
a0:64:de:3d:c8:26:59:cb:4a:46:b9:1d:17:04:7d:bf root@evs-node
```

3. **EVS ノード内の `/root/.ssh/id_rsa.pub` ファイルにある公開鍵を EVS コントローラ内の `/var/user/evsuser/.ssh/authorized_keys` ファイルにコピーします。**

4. **EVS ノードから `evsuser` として EVS コントローラにログインして、SSH 認証が設定されているかどうかを確認します。**

```
evs-node# ssh evsuser@evs-controller
The authenticity of host 'evs-controller (192.168.100.10)' can't be established.
RSA key fingerprint is 73:66:81:15:0d:49:46:e0:1d:73:32:77:4f:7c:24:a5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'evs-controller' (RSA) to the list of known hosts.
Last login: Wed Jun 11 14:36:28 2014 from evs-controller
Oracle Corporation      SunOS 5.11      11.2    April 2014
evsuser@evs-controller$
```

   この出力は、EVS ノードから `evsuser` として EVS コントローラにパスワードなしでログインできることを示しています。

▼ EVS マネージャーと EVS コントローラの間に SSH 認証を設定する方法

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **EVS マネージャーで RSA 鍵ペアを生成します。**

```
evs-manager# ssh-keygen -t rsa
```

```
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
a0:64:de:3d:c8:26:59:cb:4a:46:b9:1d:17:04:7e:bf root@evs-manager
```

3. **EVS マネージャー内の `/root/.ssh/id_rsa.pub` ファイルにある公開鍵を EVS コントローラ内の `/var/user/evsuser/.ssh/authorized_keys` ファイルにコピーします。**

4. **EVS マネージャーから evsuser として EVS コントローラにログインして、SSH 認証が設定されているかどうかを確認します。**

```
evs-manager# ssh evsuser@evs-controller
The authenticity of host 'evs-controller (192.168.100.10)' can't be established.
RSA key fingerprint is 73:66:81:15:0d:49:46:e0:1d:73:32:77:4f:7c:24:a5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'evs-controller' (RSA) to the list of known hosts.
Last login: Wed Jun 11 14:38:28 2014 from evs-controller
Oracle Corporation      SunOS 5.11      11.2    April 2014
evsuser@evs-controller$
```

この出力は、EVS マネージャーから evsuser として EVS コントローラにパスワードなしでログインできることを示しています。

▼ EVS コントローラと EVS ノードの間に SSH 認証を設定する方法

1. **管理者になります。**

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **EVS コントローラでユーザー evsuser になります。**

```
evs-controller# su - evsuser
```

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

3. **EVS コントローラで evsuser の RSA 鍵ペアを生成します。**

```
evsuser@evs-controller$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/var/user/evsuser/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

```
Your identification has been saved in /var/user/evsuser/.ssh/id_rsa.
Your public key has been saved in /var/user/evsuser/.ssh/id_rsa.pub.
The key fingerprint is:
a0:64:de:3d:c8:26:59:cb:4a:46:b9:1e:17:04:7d:bf evsuser@evs-controller
```

4. **EVS コントローラ内の /var/user/evsuser/.ssh/id_rsa.pub ファイルにある公開鍵を EVS ノード内の /var/user/evsuser/.ssh/authorized_keys ファイルにコピーします。**

5. **EVS コントローラから evsuser として EVS ノードにログインして、SSH 認証が設定されているかどうかを確認します。**

```
evsuser@evs-controller$ ssh evsuser@evs-node
The authenticity of host 'evs-node (192.168.100.20)' can't be established.
RSA key fingerprint is 73:66:89:15:0d:49:46:e0:1d:73:32:77:4f:7c:24:a5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'evs-node' (RSA) to the list of known hosts.
Last login: Wed Jun 11 14:40:28 2014 from evs-node
Oracle Corporation      SunOS 5.11      11.2    April 2014
evsuser@evs-node$
```

この出力は、EVS コントローラから evsuser として EVS ノードにパスワードなしでログインできることを示しています。

注意 - EVS の設定中に SSH 認証を設定しないと、evsadm コマンドは EVS コントローラと非対話形式で、セキュアに通信できません。

▼ EVS コントローラを構成する方法

始める前に evsadm コマンドを実行するホストと EVS コントローラの間に事前共有鍵を使用した SSH 認証を設定します。

1. **Elastic Virtual Switch Administration 権利プロファイルを持つ管理者またはユーザーになります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **EVS コントローラを設定します。**

```
# evsadm set-prop -p controller=[value[...,]]
```

このコマンドは、コマンドが実行されるホストのプロパティーの値を設定します。サポートされている唯一のプロパティーは controller であり、ssh://[*user*@]evs-controller-*host-name* または ssh://[*user*@]evs-controller-*IP-address* の形式で指定できます。

3. **(オプション) 構成した EVS コントローラを表示します。**

   ```
   # evsadm show-prop [[-c] -o field[,...]] [-p controller[,...]]
   ```

   詳細は、123 ページの「EVS コントローラの表示」を参照してください。

4. **EVS コントローラのプロパティーを設定します。**

   ```
   # evsadm set-controlprop [-h host] -p prop=[value[...,]]
   ```

   詳細は、124 ページの「EVS コントローラのプロパティーの設定」を参照してください。

5. **(オプション) EVS コントローラのプロパティーを表示します。**

   ```
   # evsadm show-controlprop [[-c] -o field[,...]] [-p prop[,...]]
   ```

   詳細は、124 ページの「EVS コントローラのプロパティーの表示」を参照してください。

例 6-1 EVS コントローラの構成

次の例は、ホスト s11-server を、L2 セグメントが VXLAN を使用して作成される EVS コントローラとして構成する方法を示します。

```
# evsadm set-prop -p controller=ssh://evsuser@s11-server
# evsadm show-prop
PROPERTY            PERM   VALUE                      DEFAULT
controller          rw     ssh://evsuser@s11-server   --
# evsadm set-controlprop -p l2-type=vxlan
# evsadm set-controlprop -p vxlan-range=10000-20000
# evsadm set-controlprop -p vxlan-addr=192.168.10.0/24
# evsadm set-controlprop -h s11-server -p uplink-port=net3
# evsadm set-controlprop -h s11-client -p uplink-port=net4
# evsadm show-controlprop
PROPERTY            PERM VALUE               DEFAULT             HOST
l2-type             rw   vxlan               vlan                --
uplink-port         rw   net3                --                  s11-server
uplink-port         rw   net4                --                  s11-client
vlan-range          rw   --                  --                  --
vlan-range-avail    r-   --                  --                  --
vxlan-addr          rw   192.168.10.0/24     0.0.0.0             --
vxlan-ipvers        rw   v4                  v4                  --
vxlan-mgroup        rw   0.0.0.0             0.0.0.0             --
vxlan-range         rw   10000-20000         --                  --
vxlan-range-avail   r-   10000-20000         --                  --
```

この例では、vxlan-range-avail プロパティーに、エラスティック仮想スイッチの実装に使用できる VXLAN ID (10000-20000) が表示されています。サブネット 192.168.10.0/24 の一部である IP インタフェースは、EVS ノードに VXLAN リンクを作成するために使用されます。

次の例は、IP アドレスが 192.168.100.1 であるホストを、L2 セグメントが VLAN を使用して作成される EVS コントローラとして構成する方法を示しています。

```
# evsadm set-prop -p controller=ssh://evsuser@192.168.100.1
# evsadm set-controlprop -p l2-type=vlan
# evsadm set-controlprop -p vlan-range=200-300,400-500
# evsadm set-controlprop -p uplink-port=net2
# evsadm set-controlprop -h host2.example.com -p uplink-port=net3
# evsadm set-controlprop -h host3.example.com -p uplink-port=net4
```

この出力は、VLAN ID 200-300 と 400-500 がエラスティック仮想スイッチ用に確保されていることを示しています。データリンク net2 は、host2.example.com と host3.example.com を除くすべてのホスト上の uplink-port です。host2 ではデータリンク net3 が uplink-port として使用され、host3 ではデータリンク net4 が uplink-port として使用されます。

例 6-2 EVS コントローラのプロパティーのリセット

次の例は、コントローラのプロパティー uplink-port をリセットする方法を示しています。

```
# evsadm show-controlprop -p uplink-port
PROPERTY            PERM     VALUE      DEFAULT             HOST
uplink-port         rw       net2       --                  --
# evsadm set-controlprop -p uplink-port=
# evsadm show-controlprop -p uplink-port
PROPERTY            PERM     VALUE      DEFAULT             HOST
uplink-port         rw       --         --                  --
```

エラスティック仮想スイッチの構成

エラスティック仮想スイッチは、1 つまたは複数の物理マシンにまたがる仮想スイッチであり、分離された L2 セグメントを表します。この分離は、VLAN または VXLAN のどちらかを使用して実装されます。EVS ノードの VNIC または anet リソースをエラスティック仮想スイッチに接続して、EVS ノード間のネットワーク接続を提供できます。詳細は、101 ページの「Oracle Solaris のエラスティック仮想スイッチ機能とは」を参照してください。

エラスティック仮想スイッチを構成する予定の場合は、仮想トポロジを理解する必要があります。必要な L2 セグメントの数、およびサブネットやデフォルトルーターを含む各ネットワークの IPnet 情報を確認します。さらに、エラスティック仮想スイッチ用に構成する必要がある仮想ポートの数や、仮想ポートに対して指定する必要があるプロパティーの決定が必要になることがあります。

エラスティック仮想スイッチの必須パッケージ

EVS クライアントおよび EVS ノードとして機能するシステムに `pkg:/service/network/evs` パッケージをインストールする必要があります。

パッケージをインストールするには、次のコマンドを使用します。

```
# pkg install evs
```

エラスティック仮想スイッチを構成するためのコマンド

このセクションでは、エラスティック仮想スイッチを構成するための次のタスクを実行する方法について説明します。

- エラスティック仮想スイッチの作成
- エラスティック仮想スイッチへの IPnet の追加
- エラスティック仮想スイッチへの VPort の追加

エラスティック仮想スイッチの作成

エラスティック仮想スイッチを作成するには、`evsadm create-evs` コマンドを使用します。コマンドの構文は次のとおりです。

```
# evsadm create-evs [-T tenant-name] [-p {prop=value[,...]}[,..]] EVS-switch-name
```

`-T` *tenant-name* テナントを指定します。テナントを指定すると、そのテナントの名前空間内にエラスティック仮想スイッチが作成されます。それ以外の場合、エラスティック仮想スイッチはデフォルトのテナント `sys-global` に作成されます。テナントは、エラスティック仮想スイッチが関連付けられているテナントを表す読み取り専用プロパティーです。

`-p` *prop* エラスティック仮想スイッチの指定された値に設定できる、コンマ区切りのプロパティーのリストを指定します。次のプロパティーを設定できます。

- `maxbw` - エラスティック仮想スイッチのポートに全二重帯域幅を設定します。この帯域幅は、スケール接尾辞 (K ビット/秒、M ビット/秒、G ビット/秒を示す `K`、`M`、`G`) が付いた整数として指定されます。単位が指定されていない場合、入力値は M ビット/秒として読み取られます。デフォルトの帯域幅の制限はありません。

- `priority` - エラスティック仮想スイッチのポートに相対的な優先順位を設定します。指定できる値は、`high`、`medium`、または `low` です。デフォルト値は `medium` です。この優先順位は、回線上のどのプロトコル優先順位フィールドでも反映されませんが、システム内のパケット処理スケジューリングに使用されます。優先順位の高い VPort では、システムリソースの可用性に応じて待機時間が短縮されます。

EVS-switch-name エラスティック仮想スイッチの名前を指定します。

エラスティック仮想スイッチを作成する方法を示す例については、例6-3「エラスティック仮想スイッチの構成」を参照してください。

エラスティック仮想スイッチへの IPnet の追加

エラスティック仮想スイッチに IPnet を追加するには、`evsadm add-ipnet` コマンドを使用します。コマンドの構文は次のとおりです。

```
# evsadm add-ipnet [-T tenant-name] -p subnet=value[{,prop=value[,...]}
[,...]]\
EVS-switch-name/IPnet-name
```

`-T` *tenant-name* テナントの名前を指定します。テナント名を指定すると、IPnet がテナントの名前空間の EVS に関連付けられます。

`-p` *prop* 特定のエラスティック仮想スイッチに設定する必要がある IPnet プロパティーのコンマ区切りのリスト。

サポートされている IPnet のプロパティーは次のとおりです。

- `subnet` - 必須。IPv4 アドレスまたは IPv6 アドレスのブロックを表します。`subnet` プロパティーは、IPnet を追加するときに指定する必要があります。そうしないと、IPnet の追加は失敗します。
- `defrouter` - オプション。特定のサブネットに対するゲートウェイの IP アドレスを指定します。`defrouter` が指定されていない場合は、範囲内の最初のアドレスがデフォルトルーターの IP アドレスとして選択されます。

EVS-switch-name/IPnet-name IPnet が関連付けられたエラスティック仮想スイッチの名前を指定します。

IPnet プロパティーの詳細は、evsadm(1M) のマニュアルページを参照してください。エラスティック仮想スイッチに IPnet を追加する方法を示す例については、例6-3「エラスティック仮想スイッチの構成」を参照してください。

エラスティック仮想スイッチへの VPort の追加

エラスティック仮想スイッチに VPort を追加するには、`evsadm add-vport` コマンドを使用します。コマンドの構文は次のとおりです。

```
# evsadm add-vport [-T tenant-name] [-p {prop=value[,...]}[,...]] EVS-switch-name/VPort-name
```

`-p` *prop*
: VPort に対して設定できる VPort プロパティーのコンマ区切りのリストを指定します。サポートされている VPort プロパティーについての詳細は、表5-1「VPort のプロパティー」を参照してください。

EVS-switch-name/VPort-name
: VPort が関連付けられたエラスティック仮想スイッチの名前を指定します。

エラスティック仮想スイッチに VPort を追加する方法を示す例については、例6-3「エラスティック仮想スイッチの構成」を参照してください。

▼ エラスティック仮想スイッチを構成する方法

始める前に エラスティック仮想スイッチを構成する計算ノード上に EVS コントローラを設定する必要があります。詳細は、130 ページの「EVS コントローラを構成する方法」の手順 2 を参照してください。

1. **Elastic Virtual Switch Administration 権利プロファイルを持つ管理者またはユーザーになります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **エラスティック仮想スイッチを作成します。**

   ```
   # evsadm create-evs [-T tenant-name] [-p {prop=value[,...]}[,..]] EVS-switch-name
   ```

   詳細は、133 ページの「エラスティック仮想スイッチの作成」を参照してください。

   注記 - 仮想ポートのプロパティーを明示的に設定すると、そのプロパティー値は対応するエラスティック仮想スイッチのプロパティー値をオーバーライドします。

   ---

3. **エラスティック仮想スイッチに IPnet を追加します。**

```
# evsadm add-ipnet [-T tenant-name] -p subnet=value[{,prop=value[,...]}[,...]]
 \
EVS-switch-name/IPnet-name
```

詳細は、134 ページの「エラスティック仮想スイッチへの IPnet の追加」を参照してください。

4. **(オプション) エラスティック仮想スイッチに VPort を追加します。**

```
# evsadm add-vport [-T tenant-name] [-p {prop=value[,...]}[,...]] EVS-switch-name/VPort-name
```

エラスティック仮想スイッチに VPort を追加すると、IPnet アドレスの範囲からランダムな MAC アドレスと IP アドレスが割り当てられます。そのため、最初に IPnet をエラスティック仮想スイッチに追加してから、VPort を追加する必要があります。`evsadm add-vport` コマンドの詳細は、135 ページの「エラスティック仮想スイッチへの VPort の追加」を参照してください。

注記 - エラスティック仮想スイッチに常に仮想ポートを追加する必要はありません。VNIC を作成するときに指定できるのは、その VNIC が接続する必要があるエラスティック仮想スイッチの名前だけです。このような場合は、EVS コントローラがシステムの仮想ポートを生成します。これらの仮想ポートは、`sys-`*vportname* という命名規則に従います (`sys-vport0` など)。システムの仮想ポートはエラスティック仮想スイッチのプロパティーを継承します。

---

5. **(オプション) 構成したエラスティック仮想スイッチを表示します。**

```
# evsadm
```

例 6-3 エラスティック仮想スイッチの構成

次の例は、エラスティック仮想スイッチ `ORA` を作成し、IPnet `ora_ipnet` を追加し、VPort `vport0` をエラスティック仮想スイッチに追加する方法を示しています。

```
# evsadm create-evs ORA
# evsadm add-ipnet -p subnet=192.168.10.0/24 ORA/ora_ipnet
# evsadm add-vport ORA/vport0
# evsadm
NAME          TENANT        STATUS VNIC          IP                HOST
ORA           sys-global    idle   --            ora_ipnet         --
   vport0     --            free   --            192.168.10.2/24   --
```

次の例は、テナント `tenantA` のエラスティック仮想スイッチ `ORA` を作成し、IPnet `ora_ipnet` を追加し、VPort `vport0` をエラスティック仮想スイッチに追加する方法を示しています。

```
# evsadm create-evs -T tenantA ORA
```

```
# evsadm add-ipnet -T tenantA -p subnet=192.168.10.0/24 ORA/ora_ipnet
# evsadm add-vport -T tenantA ORA/vport0
# evsadm
NAME          TENANT        STATUS  VNIC         IP               HOST
ORA           tenantA       idle    --           ora_ipnet        --
   vport0     --            free    --           192.168.10.2/24  --
```

エラスティック仮想スイッチ用の VNIC の作成

dladm および zonecfg コマンドを使用して、エラスティック仮想スイッチ用の VNIC を作成できるようになりました。

▼ エラスティック仮想スイッチの VNIC を作成する方法

始める前に evsadm set-prop コマンドを使用して、EVS ノードに controller プロパティーを設定する必要があります。詳細は、130 ページの「EVS コントローラを構成する方法」を参照してください。

1. **Elastic Virtual Switch Administration 権利プロファイルを持つ管理者またはユーザーになります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **エラスティック仮想スイッチの VNIC を構成します。**

   # dladm create-vnic -t -c *EVS-switch-name*[/*VPort-name*] [**-T** *tenant-name*] *VNIC-name*

-t	VNIC を一時的なものとして指定します。
-c *EVS-switch-name*[/*VPort-name*]	VNIC を接続する必要があるエラスティック仮想スイッチの名前を指定します。VPort の名前を指定すると、VNIC が指定された VPort に接続されます。VPort 名を指定しない場合は、システムが自動的に VPort を生成し、VNIC に VPort を割り当てます。エラスティック仮想スイッチに VNIC が接続されると、VNIC は指定されたエラスティック仮想スイッチまたは VPort のいずれかからプロパティーを継承します。
-T *tenant-name*	エラスティック仮想スイッチを所有するテナントの名前を指定します。テナントを指定しないと、システムはデフォルトの sys-global テナントを仮定します。
VNIC-name	VNIC の名前。

3. **(オプション) エラスティック仮想スイッチに接続された VNIC に関する情報を表示します。**

```
# dladm show-vnic -c
```

-c オプションは、エラスティック仮想スイッチに接続されている VNIC に関する情報を表示します。

例 6-4 エラスティック仮想スイッチ用の VNIC の作成

この例は、一時的な VNIC vnic1 を作成し、エラスティック仮想スイッチ ORA と VPort vport0 に VNIC を接続する方法を示しています。

```
# dladm create-vnic -t -c ORA/vport0 vnic1
# dladm show-vnic -c
LINK     TENANT       EVS     VPORT      OVER            MACADDRESS        VIDS
vnic1    sys-global   ORA     vport0     evs-vxlan10000  2:8:20:b0:6e:63   0
```

エラスティック仮想スイッチの VNIC anet リソースの作成

拡張された zonecfg コマンドを使用すると、エラスティック仮想スイッチのゾーンの VNIC anet リソースを構成できます。

ゾーンを構成する際は、anet リソースに次のプロパティーを設定できます。

- tenant – テナントの名前を指定します。ゾーンを構成するときに値を指定しないと、システムがデフォルト値、sys-global テナントを割り当てます。
- vport – VPort の名前を指定します。ゾーンを構成するときに値を指定しないと、システム VPort が自動的にエラスティック仮想スイッチに対して生成され、VPort はエラスティック仮想スイッチのプロパティーを継承します。
- evs – anet リソースを接続する必要のあるエラスティック仮想スイッチの名前を指定します。

データセンター内の VPort は、テナント名、エラスティック仮想スイッチ名、および VPort 名によって一意に識別されます。詳細は、『Oracle Solaris ゾーンの作成と使用 』を参照してください。

例 6-5 エラスティック仮想スイッチの VNIC anet リソースの作成

この例は、テナント tenantA の ORA および vport0 に接続されている、VNIC anet リソース evszone/net1 を含むゾーンを作成する方法を示しています。

```
# zonecfg -z evszone
Use 'create' to begin configuring a new zone
zonecfg:evszone> create
create: Using system default template 'SYSdefault'
zonecfg:evszone> set zonepath=/export/zones/evszone
zonecfg:evszone> set tenant=tenantA
zonecfg:evszone> add anet
zonecfg:evszone:net> set evs=ORA
zonecfg:evszone:net> set vport=vport0
zonecfg:evszone:net> end
zonecfg:evszone> exit
# zoneadm -z evszone install
# zoneadm -z evszone boot
# zlogin -C evszone
# dladm show-vnic -c
LINK          TENANT   EVS  VPORT   OVER  MACADDRESS       VIDS
evszone/net1  tenantA  ORA  vport0  net2  2:8:20:89:a1:97  200
```

`evszone` がブートするときに、VNIC `anet` `evszone/net1` が VPort `ORA/vport0` の MAC アドレス、IP アドレス、および SLA プロパティーに関連付けられます。エラスティック仮想スイッチのゾーンの VNIC `anet` リソースの構成の詳細は、154 ページの「ユースケース: エラスティック仮想スイッチの構成」を参照してください。

エラスティック仮想スイッチ、IPnet、および VPort の管理

このセクションでは、エラスティック仮想スイッチ、IPnet、および VPort を管理する方法について説明します。エラスティック仮想スイッチ、IPnet、および VPort を構成する方法の詳細は、132 ページの「エラスティック仮想スイッチの構成」を参照してください。

エラスティック仮想スイッチの管理

このセクションでは、エラスティック仮想スイッチに対して次のタスクを実行する方法について説明します。

- エラスティック仮想スイッチに関する情報の表示
- エラスティック仮想スイッチのプロパティーの設定
- エラスティック仮想スイッチのプロパティーの表示

エラスティック仮想スイッチ情報の表示

エラスティック仮想スイッチの情報を表示するには、`evsadm show-evs` コマンドを使用します。コマンドの構文は次のとおりです。

```
# evsadm show-evs [-f {fname=value[,...]}[,...]] [-L] [[-c] -o field[,...]] [EVS-switch-name]
```

`-f` {*fname*=*value*[,...]}[,...]
: 出力をフィルタ処理するために使用される、コンマで区切られた名前と値のペア (行の選択)。複数のフィルタが指定されている場合、表示される出力は、フィルタ間での AND 演算の結果です。フィルタ値が複数値である場合、表示される出力は、フィルタ値間での OR 演算の結果です。サポートされるフィルタは次のとおりです。

- `tenant`
- `evs`
- `host`
- `ipnet`
- `vport`

`-L`
: エラスティック仮想スイッチに関連付けられた VLAN ID または VXLAN セグメント ID を表示します。

`-o` *field*[,...]
: 表示する出力フィールドのコンマ区切りのリストを指定します (大文字と小文字は区別されません)。出力に列として表示される次のフィールドを指定できます。

`all`	すべての出力フィールドを表示します。
`EVS`	エラスティック仮想スイッチの名前。
`TENANT`	エラスティック仮想スイッチを所有するテナントの名前。
`STATUS`	エラスティック仮想スイッチのステータス (アイドルまたはビジー)。VNIC が接続されている VPort が 1 つ以上ある場合、エラスティック仮想スイッチはビジーです。
`NVPORTS`	エラスティック仮想スイッチに関連付けられた仮想ポートの数。
`IPNETS`	EVS に関連付けられた IP ネットワークのリスト。現在、エラスティック仮想スイッチに関連付けることができる IP ネットワークは 1 つだけです。

HOST	エラスティック仮想スイッチが複数のサーバーにまたがるホストのリスト。

例 6-6 エラスティック仮想スイッチ情報の表示

次の例では、エラスティック仮想スイッチ ORA の情報が表示されます。

```
# evsadm show-evs ORA
EVS          TENANT        STATUS NVPORTS IPNETS      HOST
ORA          sys-global    busy   1       ora_ipnet   s11-client
```

次の例では、エラスティック仮想スイッチ ORA に関連付けられた VLAN ID が表示されます。

```
# evsadm show-evs -L
EVS          TENANT        VID  VNI
ORA          tenantA       200  --
```

出力には次の情報が表示されます。

EVS	エラスティック仮想スイッチの名前
TENANT	エラスティック仮想スイッチを所有するテナントの名前
VID	エラスティック仮想スイッチの実装に使用される VLAN ID
VNI	エラスティック仮想スイッチの実装に使用される VXLAN セグメント ID

エラスティック仮想スイッチのプロパティーの設定

エラスティック仮想スイッチのプロパティーを設定するには、evsadm set-evsprop コマンドを使用します。コマンドの構文は次のとおりです。

evsadm set-evsprop [-T *tenant-name*] -p *prop*=*value*[,...] *EVS-switch-name*

-p *prop*

指定したエラスティック仮想スイッチのプロパティーの値を設定します。EVS は次のプロパティーをサポートします。

- maxbw - 指定したエラスティック仮想スイッチに接続するすべての仮想ポートの全二重帯域幅を設定します。この帯域幅は、スケール接尾辞 (K ビット/秒、M ビット/秒、G ビット/秒を示す K、M、G) が付いた整数として指定されます。単位が指定されていない場合、入力値は

M ビット/秒として読み取られます。デフォルトでは、帯域幅の制限はありません。

- priority - 指定したエラスティック仮想スイッチに接続するすべての仮想ポートのデフォルトの優先度を設定します。指定できる値は、high、medium、または low です。デフォルト値は medium です。この優先順位は、回線上のどのプロトコル優先順位フィールドでも反映されませんが、システム内のパケット処理スケジューリングに使用されます。優先順位の高い VPort では、システムリソースの可用性に応じて待機時間が短縮されます。

例 6-7 エラスティック仮想スイッチのプロパティーの設定

この例は、エラスティック仮想スイッチ ORA のプロパティーを設定する方法を示しています。

```
# evsadm set-evsprop -p maxbw=200 ORA
# evsadm set-evsprop -p priority=high ORA
```

エラスティック仮想スイッチのプロパティーの表示

エラスティック仮想スイッチのプロパティーを表示するには、evsadm show-evsprop コマンドを使用します。コマンドの構文は次のとおりです。

```
# evsadm show-evsprop [-f {fname=value[,...]}[,...] [[-c] -o field[,...]] \
[-p prop[,...]] [EVS-switch-name]
```

-f {*fname*=*value*[,...]}[,...]
: 出力をフィルタ処理するために使用される、コンマで区切られた名前と値のペア (行の選択)。複数のフィルタが指定されている場合、表示される出力は、フィルタ間での AND 演算の結果です。フィルタ値が複数値である場合、表示される出力は、フィルタ値間での OR 演算の結果です。サポートされるフィルタは次のとおりです。

- tenant – エラスティック仮想スイッチのプロパティーをテナント名でフィルタ処理します
- evs – エラスティック仮想スイッチのプロパティーをエラスティック仮想スイッチ名でフィルタ処理します
- host – エラスティック仮想スイッチのプロパティーをホスト名でフィルタ処理します

例6-8「エラスティック仮想スイッチのプロパティーの表示」は、フィルタ値に基づいた出力を示しています。

-o *field*[,...] 表示する出力フィールドのコンマ区切りのリストを指定します (大文字と小文字は区別されません)。出力に列として表示される次のフィールドを指定できます。

`all`	すべての出力フィールドを表示します。
`EVS`	エラスティック仮想スイッチの名前。
`TENANT`	エラスティック仮想スイッチを所有するテナントの名前。
`PROPERTY`	エラスティック仮想スイッチのプロパティーの名前。
`PERM`	プロパティーの読み取り権または書き込み権。表示される値は、`r-` または `rw` のいずれかです。
`VALUE`	現在のプロパティー値。値が設定されていない場合は、`--` と表示されます。値が不明な場合は、`?` と表示されます。
`DEFAULT`	プロパティーのデフォルト値。プロパティーにデフォルト値がない場合は、`--` が表示されます。
`POSSIBLE`	プロパティーに指定可能な値のコンマ区切りのリスト。指定可能な値が不明または無限の場合は、`--` が表示されます。

例 6-8 エラスティック仮想スイッチのプロパティーの表示

次の例では、エラスティック仮想スイッチ ORA 用に構成されたプロパティーが表示されます。

```
# evsadm show-evsprop ORA
EVS      TENANT        PROPERTY  PERM VALUE     DEFAULT   POSSIBLE
ORA      sys-global    maxbw     rw   200       --        --
ORA      sys-global    priority  rw   high      medium    low,medium,high
ORA      sys-global    tenant    r-   --        --        --
```

次の例では、エラスティック仮想スイッチ HR および ORA の出力が表示されます。この例では、evs フィルタを指定して、エラスティック仮想スイッチ HR および ORA の出力を取得します。

```
# evsadm show-evsprop -f evs=HR,ORA
EVS         TENANT        PROPERTY  PERM VALUE     DEFAULT   POSSIBLE
HR          tenantA       maxbw     rw   300       --        --
HR          tenantA       priority  rw   --        medium    low,medium,high
HR          tenantA       tenant    r-   --        --        --
ORA         sys-global    maxbw     rw   --        --        --
```

```
ORA         sys-global    priority  rw   --         medium    low,medium,high
ORA         sys-global    tenant    r-   --         --        --
```

IPnet 構成の管理

このセクションでは、エラスティック仮想スイッチ用の IPnet を追加したあとで、IPnet に対して次のタスクを実行する方法について説明します。

- エラスティック仮想スイッチ用に構成された IPnet の削除
- IPnet に関する情報の表示

IPnet の削除

エラスティック仮想スイッチ用に構成された IPnet を削除するには、`evsadm remove-ipnet` コマンドを使用します。コマンドの構文は次のとおりです。

evsadm remove-ipnet [-T *tenant-name*] *EVS-switch-name*/*IPnet-name*

このコマンドは、指定されたエラスティック仮想スイッチから指定された IPnet を削除します。VPort のいずれかが使用中の場合、IPnet を削除することはできません。VPort が使用中になるのは、それに VNIC が接続されている場合です。

例 6-9 エラスティック仮想スイッチ用に構成された IPnet の削除

この例は、エラスティック仮想スイッチ `ORA` から IPnet `ora_ipnet` を削除する方法を示しています。

```
# evsadm remove-ipnet ORA/ora_ipnet
```

IPnet の表示

EVS コントローラによって管理された IPnet、または指定された IPnet に関する情報を表示するには、`evsadm show-ipnet` コマンドを使用します。コマンドの構文は次のとおりです。

evsadm show-ipnet [-f {*fname*=*value*[,...]}[,...]] [[-c] -o *field*[,...]] [*IPnet-name*]

-f {*fname*=*value*[,...]} [,...] 出力をフィルタ処理するために使用される、コンマで区切られた名前と値のペア (行の選択)。複数のフィルタが指定されている場合、表示される出力は、フィルタ間での AND 演算の結果です。フィルタ値が複数

値である場合、表示される出力は、フィルタ値間での OR 演算の結果です。サポートされているフィルタは、tenant、evs、ipnet、および host です。

-o *field*[,...]

表示する出力フィールドのコンマ区切りのリストを指定します (大文字と小文字は区別されません)。出力に列として表示される次のフィールドを指定できます。

all	すべての出力フィールドを表示します。
NAME	関連付けられているエラスティック仮想スイッチの名前付きの、IPnet の名前。
IPNET	IPnet の名前。
EVS	エラスティック仮想スイッチの名前。
TENANT	エラスティック仮想スイッチを所有するテナントの名前。
SUBNET	この IPnet のサブネット (IPv4 または IPv6 のどちらか) を表します。
START	IP アドレス範囲の開始アドレス。
END	IP アドレス範囲の終了アドレス。
DEFROUTER	指定された IPnet のデフォルトルーターの IP アドレス。
AVAILRANGE	VPort に割り当てることのできる使用可能な IP アドレスのコンマ区切りリスト。

例 6-10 エラスティック仮想スイッチの IPnet の表示

この例では、エラスティック仮想スイッチ ORA 用に構成された IPnet が表示されます。

```
# evsadm show-ipnet
NAME          TENANT     SUBNET          DEFROUTER    AVAILRANGE
ORA/ora_ipnet sys-global 192.168.10.0/24 192.168.10.1 192.168.10.3-192.168.10.254
```

VPort 構成の管理

このセクションでは、VPort に対して次のタスクを実行する方法について説明します。

- VPort のプロパティーの設定
- VPort に関連付けられたプロパティーの表示
- VPort に関する情報の表示
- VPort のリセット
- VPort の削除

VPort のプロパティーの設定

VPort のプロパティーを設定するには、`evsadm set-vportprop` コマンドを使用します。コマンドの構文は次のとおりです。

```
# evsadm set-vportprop [-T tenant-name] -p prop=value[,...] EVS-switch-name/VPort-name
```

-T *tenant-name*　　テナントの名前を指定します。

-p *prop=value*[...,]　　指定した VPort のプロパティーの値を指定します。VPort に VNIC が接続されている場合は、その VPort のプロパティーを設定すると VNIC のプロパティーが変更されます。VPort プロパティーの詳細は、表 5-1「VPort のプロパティー」を参照してください。

注記 - システム VPort のプロパティーを変更することはできません。システム VPort についての詳細は、135 ページの「エラスティック仮想スイッチを構成する方法」を参照してください。

EVS-switch-name/VPort-name　　プロパティーが設定されているエラスティック仮想スイッチまたは VPort の名前を指定します。

注記 - VPort の作成後に `ipaddr`、`macaddr`、`evs`、および `tenant` プロパティーを変更することはできません。

例 6-11　　VPort のプロパティーの設定

この例は、`HR/vport0` の最大帯域幅のプロパティーを `1G` に設定する方法を示しています。

```
# evsadm set-vportprop -p maxbw=1G HR/vport0
```

VPort のプロパティーの表示

VPort のプロパティーを表示するには、`evsadm show-vportprop` コマンドを使用します。コマンドの構文は次のとおりです。

```
# evsadm show-vportprop [-f {fname=value[,...]}[,...] [[-c] -o field[,...]] \
[-p prop[,...]] [[EVS-switch-name]/[VPort-name]]
```

このコマンドは、すべての VPort または指定された VPort のいずれかの、1 つ以上のプロパティーの現在の値を表示します。VPort プロパティーが指定されていない場合は、使用可能なすべての VPort プロパティーが表示されます。VPort のプロパティーの詳細は、表5-1「VPort のプロパティー」を参照してください。

[-f {*fname*=*value*[,...]}[,...]]
: 出力をフィルタ処理するために使用される、コンマで区切られた名前と値のペア (行の選択)。複数のフィルタが指定されている場合、表示される出力は、フィルタ間での AND 演算の結果です。フィルタ値が複数値である場合、表示される出力は、フィルタ値間での OR 演算の結果です。サポートされるフィルタは次のとおりです。

- `tenant` – VPort のプロパティーをテナント名でフィルタ処理します
- `EVS` – VPort のプロパティーをエラスティック仮想スイッチ名でフィルタ処理します
- `vport` – VPort のプロパティーを VPort 名でフィルタ処理します
- `host` – VPort のプロパティーをホスト名でフィルタ処理します

-o *field*[,...]
: 表示する出力フィールドのコンマ区切りのリストを指定します (大文字と小文字は区別されません)。出力に列として表示される次のフィールドを指定できます。

`all`	すべての出力フィールドを表示します。
`NAME`	VPort が *EVS-switch-name*/*VPort-name* の形式で関連付けられているエラスティック仮想スイッチの名前を持つ VPort の名前。
`TENANT`	エラスティック仮想スイッチを所有するテナントの名前。
`PROPERTY`	VPort プロパティーの名前。
`PERM`	プロパティーの読み取り権または書き込み権。表示される値は、`r-` または `rw` のいずれかです。
`VALUE`	現在のプロパティー値。値が設定されていない場合は、`--` と表示されます。値が不明な場合は、`?` と表示されます。
`DEFAULT`	プロパティーのデフォルト値。プロパティーにデフォルト値がない場合は、`--` が表示されます。

POSSIBLE — プロパティーに指定可能な値のコンマ区切りのリスト。値に数値の範囲がある場合は、短縮形としてmin-maxが表示される可能性があります。指定可能な値が不明または無限の場合は、--が表示されます。

例 6-12 VPort のプロパティーの表示

この例では、VPort `vport0` の VPort のプロパティーが表示されます。

```
# evsadm show-vportprop ORA/vport0
NAME          TENANT      PROPERTY  PERM VALUE           DEFAULT    POSSIBLE
ORA/vport0    sys-global  cos       rw   --               0         0-7
ORA/vport0    sys-global  maxbw     rw   --               --        --
ORA/vport0    sys-global  priority  rw   --               medium    low,medium,high
ORA/vport0    sys-global  ipaddr    r-   192.168.10.2/24 --         --
ORA/vport0    sys-global  macaddr   r-   2:8:20:b0:6e:63 --         --
ORA/vport0    sys-global  evs       r-   ORA              --        --
ORA/vport0    sys-global  tenant    r-   sys-global       --        --
```

VPort の表示

VPort を表示するには、`evsadm show-vport` コマンドを使用します。コマンドの構文は次のとおりです。

```
# evsadm show-vport [-f {fname=value[,...]}[,...]] [[-c] -o field[,...]] \
[[EVS-switch-name/][VPort-name]]
```

`-f` *{fname=value[,...]}[,...]* — 出力をフィルタ処理するために使用される、コンマで区切られた名前と値のペア (行の選択)。複数のフィルタが指定されている場合、表示される出力は、フィルタ間での AND 演算の結果です。フィルタ値が複数値である場合、表示される出力は、フィルタ値間での OR 演算の結果です。サポートされるフィルタは次のとおりです。

- `tenant` – VPort のリストをテナント名でフィルタ処理します
- `EVS` – VPort のリストをエラスティック仮想スイッチ名でフィルタ処理します
- `vport` – VPort のリストを VPort 名でフィルタ処理します
- `host` – VPort のリストをホスト名でフィルタ処理します

`-o` *field[,...]* — 表示する出力フィールドのコンマ区切りのリストを指定します (大文字と小文字は区別されません)。出力に列として表示される次のフィールドを指定できます。

`all`	すべての出力フィールドを表示します。
`NAME`	*EVS-switch-name*/*VPort-name* の形式で関連付けられているエラスティック仮想スイッチの名前を持つ VPort の名前。
`TENANT`	エラスティック仮想スイッチを所有するテナントの名前。
`STATUS`	VPort が使用中か開放されているかを表示します。VPort が VNIC に関連付けられている場合、VPort は使用中です。それ以外の場合、VPort は開放されています。
`VNIC`	この VPort に関連付けられた VNIC の名前。
`HOST`	この VPort に関連付けられた VNIC が存在するホストの名前。

例 6-13 VPort 情報の表示

この例では、VPort `vport0` に関する情報が表示されます。

```
# evsadm show-vport
NAME            TENANT         STATUS VNIC       HOST
ORA/vport0      sys-global     used   vnic1      s11-client
```

VPort のリセット

VPort に関連付けられた VNIC を削除すると、その VPort の状態は `free` になります。VPort が次の状況にあると、その VPort に関連付けられた VNIC を削除しても VPort が `used` 状態になる場合があります。

- EVS ノード内の VNIC を削除したときに、EVS ノードが EVS コントローラにアクセスできない。
- VPort に関連付けられた VNIC が、EVS ノードをリブートする前に削除されない。

VPort の状態を `free` にリセットするには、`evsadm reset-vport` コマンドを使用します。コマンドの構文は次のとおりです。

```
# evsadm reset-vport [-T tenant-name] EVS-switch-name/VPort-name
```

VPort の削除

VNIC が VPort に関連付けられている場合、VPort の削除は失敗します。そのため、まず `evsadm show-vport` コマンドを使用して、削除する VPort に VNIC が関連付けられているかどうかを確認する必要があります。エラスティック仮想スイッチから VPort を削除するには、`evsadm remove-vport` コマンドを使用します。コマンドの構文は次のとおりです。

```
# evsadm remove-vport [-T tenant-name] EVS-switch-name/VPort-name
```

このコマンドでは指定された VPort を削除します。VPort が削除されると、その VPort に関連付けられた IP アドレスと MAC アドレスが解放されます。

例 6-14 VPort の削除

この例は、エラスティック仮想スイッチ `ORA` 用に構成された VPort `vport0` を削除する方法を示しています。

```
# evsadm remove-vport -T tenantA ORA/vport0
```

エラスティック仮想スイッチの削除

このセクションでは、エラスティック仮想スイッチを削除する方法について説明します。エラスティック仮想スイッチを削除できるのは、エラスティック仮想スイッチのすべての VPort が開放されている場合のみです。したがって、VPort を VNIC に関連付けることはできません。

▼ エラスティック仮想スイッチを削除する方法

1. **Elastic Virtual Switch Administration 権利プロファイルを持つ管理者またはユーザーになります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **VPort がエラスティック仮想スイッチで使用されるかどうかを確認します。**

   ```
   # evsadm show-evs
   ```

   You cannot delete an elastic virtual switch if a VPort is in use. VNIC が VPort に接続されている場合、VPort は使用中です。`evsadm show-evs` コマンド出力の `STATUS` フィールドには、エラスティック仮想スイッチがビジーかアイドルかが表示されます。

VPort が使用中の場合は、VPort に関連付けられた VNIC を次のように削除する必要があります。

```
# dladm delete-vnic VNIC
```

3. **エラスティック仮想スイッチを削除します。**

```
# evsadm delete-evs [-T tenant-name] EVS-switch-name
```

このコマンドは、指定されたエラスティック仮想スイッチと、そのエラスティック仮想スイッチに関連付けられたすべての VPort および IPnet を削除します。

例 6-15 エラスティック仮想スイッチの削除

次の例は、エラスティック仮想スイッチ `ORA` を削除する方法を示しています。

```
# evsadm show-evs
EVS           TENANT        STATUS NVPORTS IPNETS      HOST
ORA           sys-global    idle   0       ora_ipnet   --
# evsadm delete-evs ORA
# evsadm show-evs ORA
evsadm: failed to show EVS(s): evs not found
```

次の例は、ビジーであるエラスティック仮想スイッチ `EVS1` を削除する方法を示しています。

```
# evsadm show-evs EVS1
EVS           TENANT        STATUS NVPORTS IPNETS      HOST
EVS1          sys-global    busy   1       evs1_ipnet  s11-server
# evsadm show-vport EVS1/vport1
NAME                TENANT        STATUS VNIC         HOST
EVS1/vport1         sys-global    used   vnic1        s11-server
# dladm delete-vnic vnic1
# evsadm show-evs EVS1
EVS           TENANT        STATUS NVPORTS IPNETS      HOST
EVS1          sys-global    idle   1       evs1_ipnet  --
# evsadm delete-evs EVS1
# evsadm show-evs EVS1
evsadm: failed to show EVS(s): evs not found
```

エラスティック仮想スイッチのモニタリング

エラスティック仮想スイッチの仮想ポートのネットワークトラフィック統計をモニターして、次の情報を取得できます。

- VM によって送受信されたネットワークトラフィックの量。これにより、その VM 上のワークロードに関する情報が提供されます。

- インバウンド (`idrops`) およびアウトバウンド (`odrops`) が中断されたパケットの数。これらの値は障害のあるネットワークに関する情報を提供します。
- 容量計画を行う際に役立つ、計算ノード上のすべての VM で送受信されるネットワークトラフィックの量。

エラスティック仮想スイッチをモニターするには、`evsstat` コマンドを使用します。`evsstat` コマンドは、エラスティック仮想スイッチの各 VPort の実行時統計情報を報告します。また、VPort に関連付けられた VNIC の統計情報も報告します。EVS と仮想ポートについての詳細は、`evsadm`(1M) のマニュアルページを参照してください。

`evsstat` コマンドは RAD (Remote Administration Daemon) クライアントであり、リモート EVS コントローラと通信してすべての `evsstat` サブコマンドを実行します。`evsstat` コマンドを使用する前に、`evsadm set-prop` コマンドを使用して、EVS コントローラの解決可能なホスト名または IP アドレスを指定する必要があります。コマンドの構文は次のとおりです。

```
# evsadm set-prop -p controller=ssh://[username@]hostname-or-IP-address
```

さらに、`evsstat` コマンドを実行するホストと EVS コントローラの間に事前共有された公開鍵を使用した SSH 認証を設定する必要があります。`evsstat` コマンドが EVS コントローラと非対話形式で、セキュアに通信できるようにするには、事前共有された公開鍵を使用した SSH 認証が必要です。詳細は、126 ページの「SSH 認証の設定」を参照してください。

`evsstat` のコマンド構文は次のとおりです。

```
# evsstat [-f {fname=value[,...]}[,...] [[-c] -o field[,...]] [-u R|K|M|G|T|P] \
[EVS-switch-name[/VPort-name]] [interval] [count]
```

EVS-switch-name
: 統計をモニターするエラスティック仮想スイッチの名前を指定します。エラスティック仮想スイッチの名が指定されていない場合は、すべてのエラスティック仮想スイッチの統計情報が表示されます。

VPort-name
: 統計をモニターする VPort の名前を指定します。指定された VPort に接続された VNIC の統計情報のみが表示されます。エラスティック仮想スイッチの名前を指定してから、VPort の名前を指定する必要があります。

`-f` {*fname*=*val*[,...]}[,...]
: 出力をフィルタ処理するために使用される、コンマで区切られた名前と値のペア (行の選択)。複数のフィルタが指定されている場合、表示される出力は、フィルタ間での AND 演算の結果です。フィルタ値が複数値である場合、表示される出力は、フィルタ値間での OR 演算の結果です。サポートされているフィルタは、`tenant`、`evs`、および `host` です。

-o *field*[,...]]
: 表示する出力フィールドのコンマ区切りのリストを指定します (大文字と小文字は区別されません)。出力に列として表示される次のフィールドを指定できます。

- `vport`
- `evs`
- `tenant`
- `vnic`
- `host`
- `ipkts`
- `rbytes`
- `opkts`
- `idrops`
- `odrops`

-u R|K|M|G|T|P
: 統計情報を表示する単位を指定します。指定しない場合、必要に応じて異なる単位が使用され、`xy.zU` 形式で統計情報が表示されます (`x`、`y`、および `z` は数値、`U` は該当する単位です)。サポートされる単位は次のとおりです。

- `R` – raw カウント
- `K` – キロビット
- `M` – メガビット
- `G` – ギガビット
- `T` – テラビット
- `P` – ペタビット

interval
: ネットワーク統計をリフレッシュする時間を秒数で指定します。

count
: 統計情報をリフレッシュする回数を指定します。間隔を指定してから、数を指定する必要があります。

例 6-16 エラスティック仮想スイッチのモニタリング

次の例では、すべてのエラスティック仮想スイッチの統計情報が表示されます。

```
# evsstat
VPORT       EVS     TENANT      IPKTS     RBYTES     OPKTS   OBYTES
sys-vport0  ORA     sys-global  101.88K   32.86M     40.16K  4.37M
sys-vport2  ORA     sys-global  4.50M     6.78G      1.38M   90.90M
```

```
sys-vport0   HR       sys-global   132.89K   12.25M      236     15.82K
sys-vport1   HR       sys-global   144.47K   13.32M      247     16.29K
```

次の例では、指定したエラスティック仮想スイッチ `evs0` の統計情報が表示されます。

```
# evsstat ORA
VPORT         EVS      TENANT       IPKTS     RBYTES    OPKTS    OBYTES
sys-vport0    ORA      sys-global   101.88K   32.86M    40.16K   4.37M
sys-vport2    ORA      sys-global   4.50M     6.78G     1.38M    90.90M
```

次の例では、指定した VPort `evs0/sys-vport2` の統計情報が表示されます。

```
# evsstat ORA/sys-vport2
VPORT        EVS     TENANT       IPKTS   RBYTES   OPKTS  OBYTES
sys-vport2   ORA     sys-global   4.50M   6.78G    1.38M  90.90M
```

次の例は、間隔値が 1 秒、カウント値が 3 の VPort の統計情報を示しています。この統計情報は 1 秒の間隔で 3 回リフレッシュされます。

```
# evsstat ORA/sys-vport2 1 3
VPORT         EVS      TENANT       IPKTS   RBYTES   OPKTS  OBYTES
sys-vport2    ORA      sys-global   4.50M   6.78G    1.38M  90.90M
sys-vport2    ORA      sys-global   4.50M   6.78G    1.38M  90.90M
sys-vport2    ORA      sys-global   4.50M   6.78G    1.38M  90.90M
```

次の例は、指定された出力フィールドの統計を示しています。

```
# evsstat -o vport,evs,vnic,host,ipkts,opkts
VPORT          EVS      VNIC      HOST    IPKTS     OPKTS
sys-vport0     ORA      vnic0     host1   101.88K   40.16K
sys-vport2     ORA      vnic0  host2   4.50M     1.38M
sys-vport0     HR       vnic1     host1   132.89K   236
sys-vport1     HR       vnic1     host2   144.47K   247
```

エラスティック仮想スイッチのユースケースの例

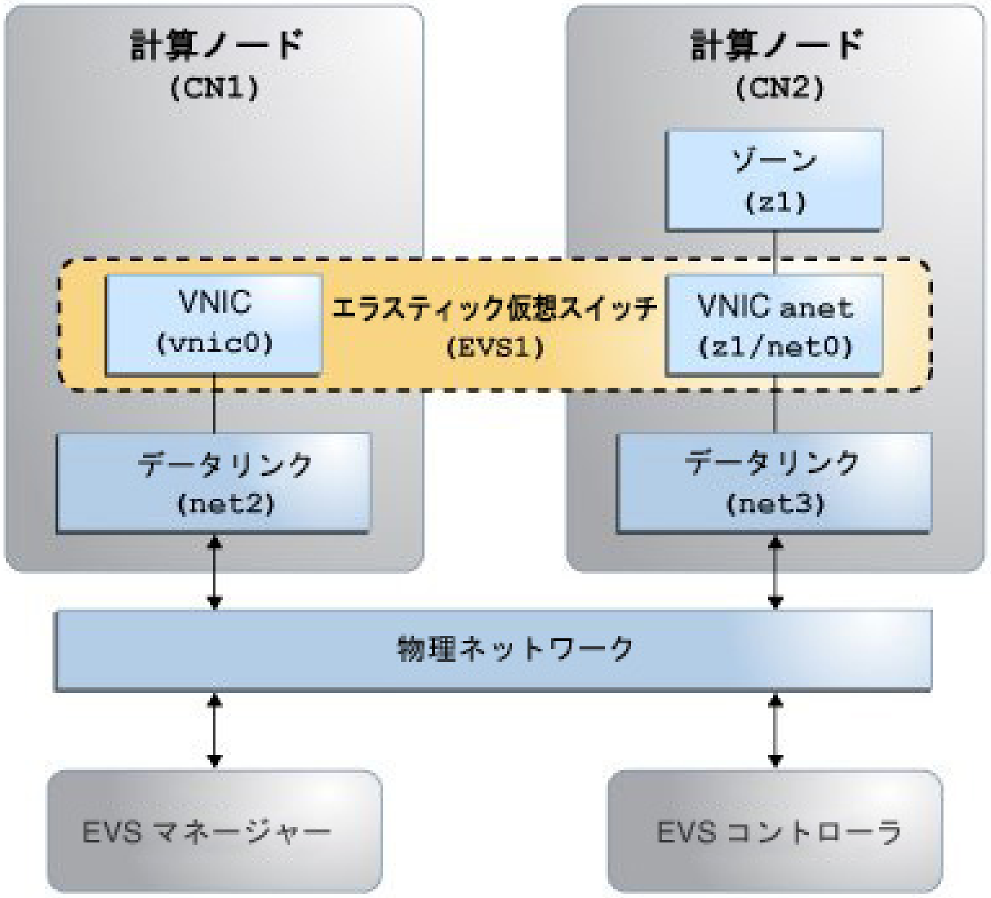
このセクションでは、エラスティック仮想スイッチを構成する方法を説明するユースケースの例を示します。

ユースケース: エラスティック仮想スイッチの構成

目的 – このユースケースは、2 つの計算ノードにまたがるエラスティック仮想スイッチ (`EVS1`) を設定する方法を示しています。

このユースケースでは、CN1 上の VNIC vnic0 とゾーン z1 の VNIC anet をエラスティック仮想スイッチ EVS1 に接続して、それらが同じ L2 セグメントの一部となり、VLAN 上で相互に通信できるようにします。次の図は、2 つの計算ノードにまたがるエラスティック仮想スイッチ (EVS1) を示しています。

図 6-2 エラスティック仮想スイッチ構成



この図は、次のコンポーネントを含む 4 つのノードがあるネットワークを示しています。

- 2 つの計算ノード (CN1 および CN2)
- VNIC anet リソースを含む CN2 上のゾーン z1 (z1/net0)

- CN1 上の VNIC vnic0
- EVS コントローラとして機能するノード (evs-controller.example.com)
- evsadm コマンドを実行する必要のある EVS マネージャーとして機能するノード (MANAGER)
- エラスティック仮想スイッチ EVS1 を実装する VLAN
- uplink-port (VLAN に使用するデータリンクを指定します)

注記 - 4 つのすべてのノードは単一のマシン上に存在できます。EVS コントローラと EVS マネージャーは同じマシン上に存在できます。

エラスティック仮想スイッチの設定の計画

1. 必須の EVS パッケージをインストールします。

   必須パッケージの詳細は、116 ページの「EVS を使用するための必須パッケージ」を参照してください。

注記 - evsuser は、pkg:/service/network/evs パッケージをインストールしたときに作成される特定のユーザーです。ユーザー evsuser には、Elastic Virtual Switch Administration 権利プロファイルが割り当てられます。このプロファイルは、EVS 操作を実行するために必要なすべての承認と特権を提供します。

---

2. EVS 設定の次のコンポーネント間に evsuser の事前共有された公開鍵を使用した SSH 認証を設定します。
   - EVS マネージャーと EVS コントローラ
   - 各 EVS ノードと EVS コントローラ
   - EVS コントローラと各 EVS ノード

   詳細は、126 ページの「SSH 認証の設定」を参照してください。

注記 - このユースケースでは、EVS ノード、EVS マネージャー、および EVS コントローラ上で controller プロパティーが ssh://evsuser@evs-controller.example.com に設定されていることを前提にしています。

---

3. EVS コントローラを構成します。

a. ネットワークで EVS コントローラとして計算ノードを指定してから、計算ノードが EVS コントローラと通信できるように各計算ノード上に EVS コントローラを設定します。コントローラのプロパティーは、その EVS コントローラと通信できる任意の計算ノードから設定できます。詳細は、132 ページの「エラスティック仮想スイッチの構成」を参照してください。

b. プロパティー `l2-type`、`vlan-range`、および `uplink-port` を指定します。それ以外の場合は、エラスティック仮想スイッチを作成できません。

4. エラスティック仮想スイッチを作成します。IPnet を関連付けて、エラスティック仮想スイッチに VPort を追加する必要があります。
5. `CN1` 上に一時的な VNIC を作成し、エラスティック仮想スイッチの VPort に VNIC を接続します。
6. ゾーン `z1` に VNIC `anet` リソースを作成し、それをエラスティック仮想スイッチに接続します。

EVS マネージャーの操作

1. EVS コントローラを設定します。

```
MANAGER# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. EVS コントローラのプロパティーを設定します。

a. エラスティック仮想スイッチに使用する必要がある L2 トポロジのタイプを設定します。

```
MANAGER# evsadm set-controlprop -p l2-type=vlan
```

b. VLAN 範囲を設定します。

```
MANAGER# evsadm set-controlprop -p vlan-range=200-300
```

c. VLAN に使用されるデータリンク (`uplink-port`) を指定します。

```
MANAGER# evsadm set-controlprop -p uplink-port=net2
```

```
MANAGER# evsadm set-controlprop -h CN2 -p uplink-port=net3
```

注記 - EVS コントローラに接続でき、必要な承認がある場合は、データセンター内の任意のノードから EVS コントローラを構成できます。詳細は、117 ページの「EVS を使用するためのセキュリティー要件」を参照してください。

---

3. コントローラのプロパティーを確認します。

```
MANAGER# evsadm show-controlprop -p l2-type,vlan-range,uplink-port
NAME                VALUE               DEFAULT             HOST
l2-type             vlan                vlan                --
vlan-range          200-300             --                  --
uplink-port         net2                --                  --
uplink-port         net3                --                  CN2
```

4. EVS1 という名前のエラスティック仮想スイッチを作成します。

```
MANAGER# evsadm create-evs EVS1
```

5. IPnet EVS1_ipnet を EVS1 に追加します。

```
MANAGER# evsadm add-ipnet -p subnet=192.168.100.0/24 EVS1/EVS1_ipnet
```

6. VPort vport0 を EVS1 に追加します。

```
MANAGER# evsadm add-vport EVS1/vport0
```

   エラスティック仮想スイッチに常に仮想ポートを追加する必要はありません。VNIC を作成するときに指定できるのは、その VNIC が接続する必要があるエラスティック仮想スイッチの名前だけです。このような場合は、EVS コントローラがシステムの仮想ポートを生成します。これらの仮想ポートは、sys-*vportname* という命名規則に従います (sys-vport0 など)。システムの仮想ポートはエラスティック仮想スイッチのプロパティーを継承します。

7. 作成されたエラスティック仮想スイッチを確認します。

```
MANAGER# evsadm
NAME           TENANT          STATUS    VNIC      IP                      HOST
EVS1           sys-global      --        --        EVS1_ipnet              --
   vport0      --              free      --        192.168.100.2/24        --
```

注記 - テナント名が指定されていないため、エラスティック仮想スイッチ EVS1 でデフォルトのテナント名 sys-global が使用されます。エラスティック仮想スイッチを作成するときに -T オプションを使用してテナント名を指定できます。詳細は、135 ページの「エラスティック仮想スイッチを構成する方法」を参照してください。

---

8. EVS1/vport0 に関連付けられた MAC アドレスと IP アドレスを確認します。

```
MANAGER# evsadm show-vportprop -p macaddr,ipaddr EVS1/vport0
```

```
NAME          TENANT        PROPERTY  PERM   VALUE             DEFAULT   POSSIBLE
EVS1/vport0   sys-global    ipaddr    r-     192.168.100.2/24  --        --
EVS1/vport0   sys-global    macaddr   r-     2:8:20:3c:78:bd   --        --
```

vport0 に接続する VNIC は IP アドレスと MAC アドレスを継承します。vport0 に割り当てられている IP アドレスは、IPnet EVS1_ipnet から次に使用できる IP アドレスであり、vport0 の MAC アドレスはランダムに生成されます。

9. エラスティック仮想スイッチ EVS1 に関連付けられた VLAN ID を確認します。

```
MANAGER# evsadm show-evs -L
EVS          TENANT          VID   VNI
EVS1         sys-global      200   --
```

計算ノード CN1 の操作

1. EVS コントローラを指定します。

```
CN1# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. 一時的な VNIC vnic0 を作成して、EVS1/vport0 に接続します。

```
CN1# dladm create-vnic -t -c EVS1/vport0 vnic0
```

3. 作成された VNIC を確認します。

```
CN1# dladm show-vnic -c
LINK     TENANT          EVS     VPORT       OVER     MACADDRESS        VIDS
vnic0    sys-global      EVS1    vport0      net2     2:8:20:3c:78:bd   200
```

vnic0 の MAC アドレスが VPort の MAC アドレスにマップされます。

4. vnic0 に使用できる IP アドレスを確認します。

```
CN1# dladm show-linkprop -p allowed-ips vnic0
LINK     PROPERTY     VALUE             EFFECTIVE        DEFAULT   POSSIBLE
vnic0    allowed-ips  192.168.100.2     192.168.100.2    --        --
```

allowed-ips プロパティーが VPort に関連付けられた IP アドレスに設定されます。この設定では、vnic0 に 192.168.100.2 以外の IP アドレスを作成することはできません。

5. vnic0 用の IP インタフェースを作成し、IP アドレスとして 192.168.100.2 を割り当てます。

```
# ipadm create-ip -t vnic0
```

```
# ipadm create-addr -t -a 192.168.100.2 vnic0
```

計算ノード CN2 の操作

1. EVS コントローラを指定します。

```
CN2# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. ゾーン z1 用の VNIC anet リソースを構成し、それをエラスティック仮想スイッチに接続します。

```
CN2# zonecfg -z z1
Use 'create' to begin configuring a new zone
zonecfg:z1> create
create: Using system default template 'SYSdefault'
zonecfg:z1> set zonepath=/export/zones/z1
zonecfg:z1> select anet linkname=net0
zonecfg:z1:anet> set evs=EVS1
zonecfg:z1:anet> end
zonecfg:z1> commit
zonecfg:z1> exit
```

3. ゾーン z1 をインストールしてブートします。

```
CN2# zoneadm -z z1 install
CN2# zoneadm -z z1 boot
```

4. ゾーン z1 にログインしてゾーン構成を完了します。

```
CN2# zlogin -C z1
```

ゾーン構成についての詳細は、『Oracle Solaris ゾーンの作成と使用 』を参照してください。

5. 作成された VNIC anet を確認します。

```
CN2# dladm show-vnic -c
LINK       TENANT       EVS    VPORT        OVER   MACADDRESS        VIDS
z1/net0    sys-global   EVS1   sys-vport0   net2   2:8:20:1a:c1:e4   200
```

VNIC anet リソースを作成したときに VPort が指定されなかったため、EVS コントローラは、VNIC anet リソース用にシステム VPort sys-vport0 を作成します。

6. VPort に関連する情報を表示します。

```
CN2# evsadm show-vport -o all
NAME            TENANT     STATUS VNIC    HOST MACADDR         IPADDR
EVS1/sys-vport0 sys-global used   z1/net0 CN2  2:8:20:1a:c1:e4 192.168.100.3/24
```

VNIC anet リソースが plumb され、VPort の IP アドレスが割り当てられます。

7. VNIC anet リソース、z1/net0 の IP アドレスを確認します。

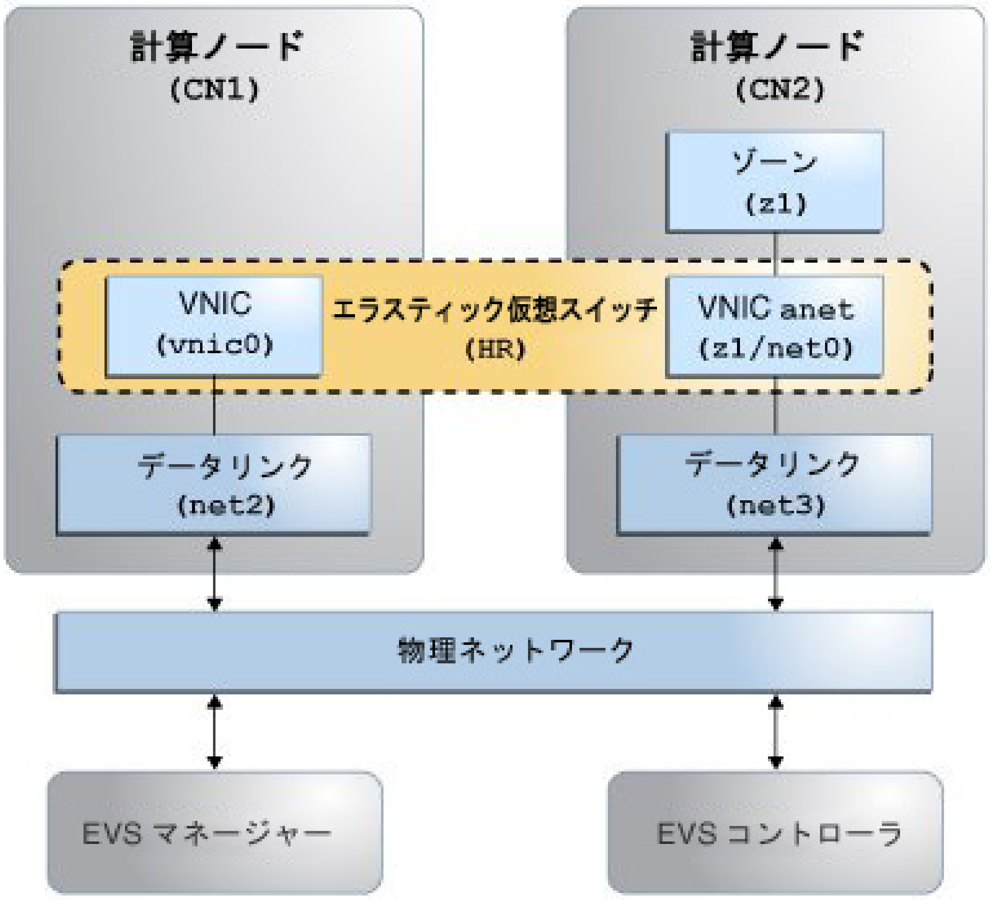
```
CN2# zlogin z1 ipadm
NAME              CLASS/TYPE    STATE       UNDER      ADDR
lo0               loopback      ok          --         --
   lo0/v4         static        ok          --         127.0.0.1/8
   lo0/v6         static        ok          --         ::1/128
net0              ip            ok          --         --
   net0/v4        inherited     ok          --         192.168.100.3/24
```

ユースケース: テナント用のエラスティック仮想スイッチの構成

目的 – このユースケースは、テナントの 2 つの計算ノードにまたがるエラスティック仮想スイッチ (HR) を設定する方法を示しています。

このユースケースでは、CN1 上の VNIC vnic0 とゾーン z1 の VNIC anet をエラスティック仮想スイッチ HR に接続して、それらが同じ L2 セグメントの一部となり、VXLAN 上で相互に通信できるようにします。VNIC はテナント tenantA の一部です。次の図は EVS の設定を示しています。

図 6-3 テナントのエラスティック仮想スイッチ構成



この図は、次のコンポーネントを含む 4 つのノードがあるネットワークを示しています。

- 2 つの計算ノード (`CN1` および `CN2`)
- VNIC `anet` リソースを含む `CN2` 上のゾーン `z1`
- `CN1` 上の VNIC `vnic0`
- EVS コントローラとして機能するノード、`CONTROLLER`
- `evsadm` コマンドを実行する必要のある EVS マネージャーとして機能するノード、`MANAGER`
- エラスティック仮想スイッチ `HR` を実装する VXLAN

- `uplink-port` (VXLAN に使用するデータリンクを指定します)

エラスティック仮想スイッチの設定の計画

1. 必須の EVS パッケージをインストールします。必須パッケージの詳細は、116 ページの「EVS を使用するための必須パッケージ」を参照してください。

注記 - `evsuser` は、`pkg:/service/network/evs` パッケージをインストールしたときに作成される特定のユーザーです。ユーザー `evsuser` には、Elastic Virtual Switch Administration 権利プロファイルが割り当てられます。このプロファイルは、EVS 操作を実行するために必要なすべての承認と特権を提供します。

---

2. EVS 設定の次のコンポーネント間に `evsuser` の事前共有された公開鍵を使用した SSH 認証を設定します。
   - EVS マネージャーと EVS コントローラ
   - 各 EVS ノードと EVS コントローラ
   - EVS コントローラと各 EVS ノード

   詳細は、126 ページの「SSH 認証の設定」を参照してください。

注記 - このユースケースでは、各 EVS ノード、EVS マネージャー、および EVS コントローラ上で `controller` プロパティーが `ssh://evsuser@evs-controller.example.com` に設定されていることを前提にしています。

---

3. EVS コントローラを構成し、コントローラのプロパティーを設定します。
   a. すべての計算ノード上で EVS コントローラを設定してから、各計算ノードにまたがるエラスティック仮想スイッチを実装する方法を指定するコントローラプロパティーを設定します。
   b. プロパティー `l2-type`、`vxlan-range`、および `uplink-port` を指定します。それ以外の場合は、エラスティック仮想スイッチを作成できません。
4. エラスティック仮想スイッチを作成します。IPnet を関連付けて、エラスティック仮想スイッチに VPort を追加する必要があります。

5. CN1 上に一時的な VNIC を作成し、エラスティック仮想スイッチの VPort に VNIC を接続します。
6. ゾーン z1 に VNIC anet を作成し、VNIC anet リソースをエラスティック仮想スイッチに接続します。

EVS マネージャーの操作

1. EVS コントローラを設定します。

```
MANAGER# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. EVS コントローラのプロパティーを設定します。
   a. エラスティック仮想スイッチに使用する必要がある L2 トポロジのタイプを設定します。この例では VXLAN を使用します。

   ```
   MANAGER# evsadm set-controlprop -p l2-type=vxlan
   ```

   b. VXLAN 範囲を設定します。

   ```
   MANAGER# evsadm set-controlprop -p vxlan-range=200-300
   ```

   c. VXLAN に使用されるデータリンク (uplink-port) を指定します。

   ```
   MANAGER# evsadm set-controlprop -p uplink-port=net2

   MANAGER# evsadm set-controlprop -h CN2 -p uplink-port=net3
   ```

注記 - EVS コントローラに接続でき、必要な承認がある場合は、データセンター内の任意のノードからコントローラを構成できます。詳細は、117 ページの「EVS を使用するためのセキュリティー要件」を参照してください。

---

3. EVS コントローラプロパティーを確認します。

```
MANAGER# evsadm show-controlprop -p l2-type,vxlan-range,uplink-port
NAME            VALUE           DEFAULT         HOST
l2-type         vxlan           vlan            --
vxlan-range     200-300         --              --
uplink-port     net2            --              --
uplink-port     net3            --              CN2
```

4. テナント tenantA のエラスティック仮想スイッチ HR を作成します。

```
MANAGER# evsadm create-evs -T tenantA HR
```

5. IPnet `hr_ipnet` をエラスティック仮想スイッチ `HR` に追加します。

```
MANAGER# evsadm add-ipnet -T tenantA -p subnet=192.168.100.0/24 HR/hr_ipnet
```

6. VPort `vport0` をエラスティック仮想スイッチ `HR` に追加します。

```
MANAGER# evsadm add-vport -T tenantA HR/vport0
```

7. テナント `tenantA` 用に作成されたエラスティック仮想スイッチを確認します。

```
MANAGER# evsadm
NAME          TENANT        STATUS VNIC         IP                HOST
HR            tenantA       --     --           hr_ipnet          --
   vport0     --            free   --           192.168.100.2/24  --
```

8. `HR/vport0` に関連付けられた MAC アドレスと IP アドレスを確認します。

```
MANAGER# evsadm show-vportprop -p macaddr,ipaddr HR/vport0
NAME          TENANT    PROPERTY  PERM VALUE            DEFAULT   POSSIBLE
HR/vport0     tenantA   ipaddr    r-   192.168.100.2/24 --        --
HR/vport0     tenantA   macaddr   r-   2:8:20:d8:da:10  --        --
```

9. エラスティック仮想スイッチ `HR` に関連付けられた VXLAN セグメント ID を確認します。

```
MANAGER# evsadm show-evs -L
EVS           TENANT        VID VNI
HR            tenantA       --  200
```

計算ノード CN1 の操作

1. EVS コントローラを指定します。

```
CN1# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. 一時的な VNIC `vnic0` を作成して、エラスティック仮想スイッチ `HR/vport0` に接続します。

```
CN1# dladm create-vnic -t -T tenantA -c HR/vport0 vnic0
```

3. 作成された VNIC を確認します。

```
CN1# dladm show-vnic -c
LINK   TENANT    EVS   VPORT    OVER           MACADDRESS        VIDS
vnic0  tenantA   HR    vport0   evs-vxlan200   2:8:20:d8:da:10   0
```

vnic0 の MAC アドレスが VPort の MAC アドレスにマップされます。

4. vnic0 に使用できる IP アドレスを確認します。

```
CN1# dladm show-linkprop -p allowed-ips vnic0
LINK     PROPERTY     VALUE         EFFECTIVE      DEFAULT   POSSIBLE
vnic0    allowed-ips  192.168.100.2 192.168.100.2  --        --
```

allowed-ips プロパティーが VPort に関連付けられた IP アドレスに設定されます。この出力は、vnic0 上に 192.168.100.2 以外の IP アドレスを作成できないことを意味します。

5. vnic0 用の IP インタフェースを作成し、IP アドレスとして 192.168.100.2 を割り当てます。

```
# ipadm create-ip -t vnic0
# ipadm create-addr -t -a 192.168.100.2 vnic0
```

6. 自動的に生成される VXLAN データリンクをチェックします。

```
CN1# dladm show-vxlan
LINK             ADDR                  VNI   MGROUP
evs-vxlan200     0.0.0.0               200   224.0.0.1
```

計算ノード CN2 の操作

1. EVS コントローラを指定します。

```
CN2# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. ゾーン z1 用の VNIC anet を構成し、それをエラスティック仮想スイッチに接続します。

```
CN2# zonecfg -z z1
Use 'create' to begin configuring a new zone
zonecfg:z1> create
create: Using system default template 'SYSdefault'
zonecfg:z1> set zonepath=/export/zones/z1
zonecfg:z1> set tenant=tenantA
zonecfg:z1> select anet linkname=net0
zonecfg:z1:anet> set evs=HR
zonecfg:z1:anet> end
zonecfg:z1> commit
zonecfg:z1> exit
```

3. ゾーン z1 をインストールしてブートします。

```
CN2# zoneadm -z z1 install
CN2# zoneadm -z z1 boot
```

4. ゾーン z1 にログインしてゾーン構成を完了します。

```
CN2# zlogin -C z1
```

ゾーン構成についての詳細は、『Oracle Solaris ゾーンの作成と使用 』を参照してください。

5. 作成された VNIC anet リソースを確認します。

```
CN2# dladm show-vnic -c
LINK      TENANT   EVS   VPORT       OVER          MACADDRESS        VIDS
z1/net0   tenantA  HR    sys-vport0  evs-vxlan200  2:8:20:1a:c1:e4   0
```

VPort が指定されていないため、EVS コントローラが VNIC anet リソース用のシステム VPort sys-vport0 を作成します。

6. VPort に関連する情報を表示します。

```
CN2# evsadm show-vport -o all
NAME            TENANT  STATUS VNIC     HOST MACADDR         IPADDR
HR/sys-vport0   tenantA used   z1/net0  CN2  2:8:20:1a:c1:e4 192.168.100.3/24
```

VNIC anet リソースが plumb され、VPort の IP アドレスが割り当てられます。

7. VNIC anet z1/net0 の IP アドレスを確認します。

```
CN2# zlogin z1 ipadm
NAME              CLASS/TYPE STATE        UNDER      ADDR
lo0               loopback   ok           --         --
   lo0/v4         static     ok           --         127.0.0.1/8
   lo0/v6         static     ok           --         ::1/128
net0              ip         ok           --         --
   net0/v4        inherited  ok           --         192.168.100.3/24
```

♦ ♦ ♦ 第 7 章

ネットワークリソースの管理

この章では、データリンクプロパティーおよびフローを使用してネットワークリソースを管理および割り当てる方法について説明します。ネットワークリソースを管理することによって、仮想ネットワークと物理ネットワークのパフォーマンスを向上させる IP サービスの品質 (QoS) を実装できます。ネットワークリソースの管理の概要については、21 ページの「ネットワークリソースの管理の概要」を参照してください。

この章の内容は、次のとおりです。

- 169 ページの「データリンクプロパティーを使用することによるネットワークリソースの管理」
- 171 ページの「NIC リングの管理」
- 181 ページの「プールと CPU の管理」
- 186 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」
- 190 ページの「ユースケース: データリンクとフローのプロパティーを設定することによるネットワークリソースの管理」

データリンクプロパティーを使用することによるネットワークリソースの管理

データリンクにネットワークリソースを割り当てて、システムのパケットの処理効率を向上させることができます。ネットワークリソースは、データリンクを作成するときにデータリンクプロパティーを設定することによって割り当てることができます。あるいは、既存のデータリンクにデータリンクプロパティーを設定できます。`dladm` コマンドを使用することによって、次のデータリンクプロパティーを設定してデータリンクにネットワークリソースを割り当てることができます。

- `maxbw` - データリンクに割り当て可能な帯域幅の最大量を指定します。詳細は、190 ページの「ユースケース: データリンクとフローのプロパティーを設定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。

- `rxrings` と `txrings` – 特定のデータリンクに割り当てることができる NIC の受信リングと送信リングの数を指定します。詳細は、171 ページの「NIC リングの管理」を参照してください。
- `pool` - ネットワークプロセスを効率的に管理するためにデータリンクに割り当てることができる CPU のセットを含む CPU プールの名前を指定します。詳細は、181 ページの「プールと CPU の管理」を参照してください。
- `cpus` - データリンクに割り当てることができる CPU の名前を指定します。詳細は、181 ページの「プールと CPU の管理」を参照してください。

Oracle Solaris でのネットワークリソースの管理のデモについては、Oracle Solaris を使用したネットワークリソースの管理 (http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/ManagingNetworkResources/ManagingNetworkResources.htm) についてのページを参照してください。

データリンクにリソースを割り当てるためのコマンド

次のコマンドは、データリンクでネットワークリソースを割り当てるために使用されます。

- 仮想リンクの作成とリソースの割り当てを同時に行うには、次のコマンド構文を使用します。

```
# dladm create-vnic -l link -p prop=value[,...] VNIC
```

link リンクの名前を示します (物理リンクまたは仮想リンクのいずれかを指定できます)。

prop データリンクプロパティーを示します。リソース割り当てのために設定できるさまざまなタイプのデータリンクプロパティーの詳細は、169 ページの「データリンクプロパティ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。

- 既存のリンクのプロパティーを設定するには、次のコマンド構文を使用します。

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

詳細については、`dladm`(1M) のマニュアルページを参照してください。

NIC リングの管理

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。ネットワークトラフィックに従ってリングを管理および割り当てることによって、システムのパケットの処理効率が向上します。たとえば、より多くのパケットを受信するリンクには、より多くの受信 (Rx) リングを割り当てることができます。

MAC クライアントへのリングの割り当て

物理データリンクや VNIC 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。MAC クライアントは、ハードウェアベースのクライアントまたはソフトウェアベースのクライアントのどちらかです。

ハードウェアベースのクライアント

1 つ以上の NIC リングを排他的に使用するクライアントは、ハードウェアベースのクライアントと呼ばれます。NIC でサポートされるリング割り当てに応じて、ハードウェアベースのクライアントで排他的に使用するリングを割り当てることができます。

ソフトウェアベースのクライアント

NIC リングを排他的に使用しないクライアントは、ソフトウェアベースのクライアントと呼ばれます。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。

VLAN でのリングの割り当て

VLAN でのリング割り当ては、その VLAN が作成された方法によって異なります。

VLAN は次の方法で作成できます。

- `dladm create-vlan` コマンドを使用して:

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

dladm create-vlan コマンドを使用して作成された VLAN は、ベースとなるデータリンクと同じ MAC アドレスを共有します。したがって、VLAN は、ベースとなるデータリンクの Rx リングと Tx リングも共有します。VLAN の構成についての詳細は、『Oracle Solaris 11.2 でのネットワークデータリンクの管理 』の「VLAN の構成」を参照してください。

- dladm create-vnic コマンドを使用して:

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

dladm create-vnic コマンドを使用して VNIC として作成された VLAN には、ベースとなるデータリンクとは異なる MAC アドレスが割り当てられます。このタイプの VLAN へのリングの割り当ては、ベースとなるデータリンクの割り当てとは独立して行われます。したがって、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、VLAN には独自の専用リングを割り当てることができます。クライアントにリングを割り当てる方法の詳細は、175 ページの「クライアントの構成とリングの割り当て」を参照してください。

リングを構成するためのコマンド

データリンクのリングを構成するには、次の dladm サブコマンドを使用します。

- # dladm show-linkprop *link*

Rx リングと Tx リングを含む、データリンクプロパティーの現在の値を表示します。例については、例7-1「データリンク上でのリングの使用とリング割り当て」 を参照してください。

次の表で、dladm show-linkprop コマンドを使用することで表示されるリングプロパティーについて説明します。

リングプロパティー	アクセス権	説明
rxringsavail	読み取り専用	物理データリンク上のハードウェアベースのクライアントに割り当てることができる Rx リングの数を示します。
rxhwclntavail	読み取り専用	物理データリンク上に作成できるハードウェアベースの Rx クライアントの数を示します。
rxrings	読み取りおよび書き込み	データリンクによって排他的に使用される Rx リングの数を示します。このプロパティーは 3 つの指定可能な値のいずれかに設定できます。

リングプロパティー	アクセス権	説明
		■ `hw` は、ハード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。この値は、ベースとなる物理リンク上のハードウェアベースの Rx クライアント (`rxhwclntavail`) がゼロより多い場合に設定できます。 ■ *number* は、データリンクに割り当てることができるリングの数を示します。この値は、ベースとなる物理リンク上の Rx リング (`rxringsavail`) がゼロより多い場合に設定できます。 ■ `sw` は、データリンクがソフトウェアベースのクライアントであることを示します。
`txringsavail`	読み取り専用	物理データリンク上のハードウェアベースのクライアントに割り当てることができる Tx リングの数を示します。
`txhwclntavail`	読み取り専用	物理データリンク上に作成できるハードウェアベースの Tx クライアントの数を示します。
`txrings`	読み取りおよび書き込み	データリンクによって排他的に使用される Tx リングの数を示します。このプロパティーは 3 つの指定可能な値のいずれかに設定できます。 ■ `hw` は、ハード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。この値は、ベースとなる物理リンク上のハードウェアベースの Tx クライアント (`txhwclntavail`) がゼロより多い場合に設定できます。 ■ *number* は、データリンクに割り当てることができるリングの数を示します。この値は、ベースとなる物理リンク上の Tx リング (`txringsavail`) がゼロより多い場合に設定できます。 ■ `sw` は、データリンクがソフトウェアベースのクライアントであることを示します。

- `# dladm show-phys -H` *link*

  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* `-l` *link VNIC*

-p *ring-properties* 値を設定できるリングプロパティーを示します。

特定の数の Rx または Tx リングを持つクライアントを作成します。

- `# dladm set-linkprop -p` *ring-properties VNIC*

  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。

データリンク上でのリングの使用とリング割り当ての表示

データリンクの Rx リングと Tx リングの指定可能な値、構成されている値、および有効値を表示するには、次のコマンド構文を使用します。

```
# dladm show-linkprop -p rxrings,txrings link
```

物理データリンクのリングが現在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するには、次のコマンド構文を使用します。

```
# dladm show-phys -H link
```

例 7-1 データリンク上でのリングの使用とリング割り当て

次の例は、データリンク `net4` のリング割り当てを示しています。

```
# dladm show-linkprop net4
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
...
net4     rxrings         rw   1            --           --        sw,hw,<1-7>
net4     txrings         rw   1            --           --        sw,hw,<1-11>
net4     txringsavail    r-   10           10           --        --
net4     rxringsavail    r-   7            7            --        --
net4     rxhwclntavail   r-   3            3            --        --
net4     txhwclntavail   r-   3            3            --        --
...
```

この出力は、データリンク `net4` が 1 つの Rx リングと 1 つの Tx リングを排他的に使用することを示しています。データリンク `net4` には、クライアントへの割り当てに使用できる 7 つの Rx リングと 10 個の Tx リングがあります。データリンク `net4` 上には、3 つのハードウェアベースの Rx クライアントと 3 つのハードウェアベースの Tx クライアントを作成できます。

次の例は、データリンク `net0` でのリングの使用を示しています。

```
# 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      --
```

この出力によると、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。

クライアントの構成とリングの割り当て

このセクションでは、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。

▼ クライアントを構成してリングを割り当てる方法

172 ページの「リングを構成するためのコマンド」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ーを表示する 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。この情報は、クライアントを構成してリングを割り当てる際に役立ちます。

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **ベースとなる物理データリンクのプロパティーを表示します。**

   ```
   # dladm show-linkprop -p rxringsavail,txringsavail,rxhwclntavail,txhwclntavail link
   ```

   コマンドの出力から次の情報を確認します。

   - 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
   - ハードウェアベースのクライアントに割り当て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

3. **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。**

   - **ハードウェアベースのクライアントは次の構文を使用して作成します。**

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

ここで、*value* は、次のいずれかを指定できます。

- hw - ハード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。
- *number* - ハードウェアベースのクライアントのみを構成していることを示します。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。

- **ソフトウェアベースのクライアントは次の構文を使用して作成します。**

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

または、クライアントが以前に作成されている場合は、dladm set-linkprop コマンドを使用してリングプロパティーを設定できます。

4. **(オプション) 作成したクライアントのリング情報を確認します。**

   # dladm show-linkprop -p rxrings,txrings *VNIC*

5. **(オプション) 異なるクライアントの間で分散されているリンクのリングを確認します。**

   # dladm show-phys -H *link*

例 7-2 nxge デバイスでのクライアントの構成とリングの割り当て

この例では、nxge デバイスを基にして、データリンク net5 上にクライアントを構成してリングを割り当てる方法を示します。この例では次のクライアントを作成する方法を示します。

- VNIC vnic2 (Rx および Tx リングを排他的に使用するハードウェアベースのクライアント)。
- VNIC vnic3 (NIC ドライバの初期構成に従って、固定された数のリングが設定されているハードウェアベースのクライアント)。
- VNIC vnic4 (ソフトウェアベースのクライアント)。

1. 物理データリンク net5 がクライアントに対するリング割り当てをサポートするかどうかを確認します。

```
# dladm show-linkprop -p rxringsavail,txringsavail net5
LINK     PROPERTY        PERM  VALUE     EFFECTIVE   DEFAULT    POSSIBLE
net5     rxringsavail    r-    7         7           --         --
net5     txringsavail    r-    11        11          --         --
```

この出力は、物理データリンク net5 に、物理データリンク net5 上のクライアントに割り当てることができる 7 個の Rx リングと 11 個の Tx リングが存在することを示しています。

2. 物理データリンク net5 上に作成できるハードウェアベースのクライアントの可用性を確認します。

```
# dladm show-linkprop -p rxhwclntavail,txhwclntavail net5
LINK     PROPERTY        PERM  VALUE    EFFECTIVE  DEFAULT   POSSIBLE
net5     rxhwclntavail   r-    3        3          --        --
net5     txhwclntavail   r-    4        4          --        --
```

この出力は、データリンク net5 上に 3 つのハードウェアベースの Rx クライアントと 4 つのハードウェアベースの Tx クライアントを作成できることを示しています。

3. 物理データリンク net5 上の既存のリングの使用状況を確認します。

```
# dladm show-phys -H net5
LINK     RINGTYPE    RINGS     CLIENTS
nxge1    RX          0-7       <default,mcast>
nxge1    TX          0-11      <default>
```

この出力は、nxge1 デバイスに 8 個の Rx リング (0-7) と 12 個の Tx リング (0-11) があることを示しています。nxge1 デバイス上にデータリンクが存在しないため、Rx リングと Tx リングはどのデータリンクにも割り当てられていません。CLIENTS 列の値 <default> は、Tx リングがソフトウェアベースのクライアントによって使用されることを示しています。CLIENTS 列の下の値 <default,mcast> は、Rx リングがソフトウェアベースのクライアントとユニキャスト以外のパケットによって使用されることを意味します。

4. データリンク net5 上に 2 つの Rx リングと 2 つの Rx リングがある VNIC vnic2 を作成します。

```
# dladm create-vnic -l net5 -p rxrings=2,txrings=2 vnic2
```

5. VNIC vnic2 に割り当てられたリングを確認します。

```
# dladm show-linkprop -p rxrings,txrings vnic2
LINK     PROPERTY    PERM  VALUE    EFFECTIVE   DEFAULT   POSSIBLE
vnic2    rxrings     rw    2        2           --        sw,hw,<1-7>
vnic2    txrings     rw    2        2           --        sw,hw,<1-11>
```

6. 物理データリンク net5 上のリングの使用状況を確認します。

```
# dladm show-phys -H net5
LINK     RINGTYPE    RINGS      CLIENTS
nxge1    RX          0,3-7      <default,mcast>
nxge1    TX          0,3-11     <default>
```

```
nxge1    RX          1-2        vnic2
nxge1    TX          1-2        vnic2
```

この出力は、vnic2 に割り当てられた Rx リングが 1 と 2 であることを示しています。Tx リングの場合、vnic2 はリング 1 および 2 を使用します。

7. 物理データリンク net5 上に追加のハードウェアベースのクライアントを作成できるかどうかを確認します。

```
# dladm show-linkprop -p rxhwclntavail,txhwclntavail net5
LINK     PROPERTY        PERM  VALUE     EFFECTIVE  DEFAULT   POSSIBLE
net5     rxhwclntavail   r-    2         2          --        --
net5     txhwclntavail   r-    3         3          --        --
```

この出力は、物理データリンク net5 上に 2 つのハードウェアベースの Rx クライアントと 3 つのハードウェアベースの Tx クライアントを作成できることを示しています。

8. VNIC vnic3 (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 -l net5 -p rxrings=hw,txrings=hw vnic3
```

9. VNIC vnic3 に割り当てられたリングを確認します。

```
# dladm show-linkprop -p rxrings,txrings vnic3
LINK     PROPERTY     PERM    VALUE   EFFECTIVE    DEFAULT   POSSIBLE
vnic3    rxrings      rw      --      1            --        sw,hw,<1-7>
vnic3    txrings      rw      hw      hw           --        sw,hw,<-11>
```

注記 - クライアントに割り当てられるリングの数はネットワークデバイスによって異なります。リングの数を明示的に指定できるデバイス (たとえば、nxge デバイス) 上のクライアントに 1 つのリングが割り当てられます。その他のデバイスの場合、クライアントに割り当てられるリングの数は、デバイスがどのように構成されているかによって異なります。例7-3「ixgbe デバイスでのクライアントの構成とリングの割り当て」を参照してください。

---

10. 物理データリンク net5 上に追加のハードウェアベースのクライアントを作成できるかどうかを確認します。

```
# dladm show-linkprop -p rxhwclntavail,txhwclntavail net5
LINK     PROPERTY        PERM   VALUE    EFFECTIVE  DEFAULT   POSSIBLE
net5     rxhwclntavail   r-     2        2          --        --
net5     txhwclntavail   r-     2        2          --        --
```

この出力は、物理データリンク net5 上に 2 つのハードウェアベースの Rx クライアントと 2 つのハードウェアベースの Tx クライアントを作成できることを示しています。

11. VNIC vnic4 (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 -l net5 -p rxrings=sw,txrings=sw vnic4
```

12. vnic4 上のリングの使用状況を確認します。

```
# dladm show-linkprop -p rxrings,txrings vnic4
LINK     PROPERTY     PERM  VALUE    EFFECTIVE   DEFAULT   POSSIBLE
vnic4    rxrings      rw    sw       --          --        sw,hw,<1-7>
vnic4    txrings      rw    sw       --          --        sw,hw,<1-11>
```

13. 物理データリンク net5 上のリングの使用状況を確認します。

```
# dladm show-phys -H net5
LINK     RINGTYPE    RINGS      CLIENTS
nxge1    RX          0,4-7      <default,mcast>,vnic4
nxge1    TX          0,4-11     <default>,vnic4
nxge1    RX          1-2        vnic2
nxge1    RX          3          vnic3
nxge1    TX          1-2        vnic2
nxge1    TX          3          vnic3
```

この出力は、vnic4 が、物理データリンク net5 でリングのデフォルトセットを共有するソフトウェアベースのクライアントであることを示しています。VNIC vnic2 は 2 つのリング (2-3) を排他的に使用するハードウェアベースのクライアントであり、vnic3 は 1 つのリング (3) を排他的に使用するハードウェアベースのクライアントです。

例 7-3 ixgbe デバイスでのクライアントの構成とリングの割り当て

この例では、ixgbe デバイスを基にして、物理データリンク net4 上にクライアントを構成してリングを割り当てる方法を示します。

1. 物理データリンク net4 上の既存のリングの使用状況を確認します。

```
# dladm show-phys -H net4
LINK      RINGTYPE    RINGS     CLIENTS
net4      RX          0-3       <default,mcast>
net4      RX          4-7       --
net4      RX          8-11      --
```

```
net4      RX          12-15     --
net4      TX          0-7       <default>
```

2. 物理データリンク net4 上にハードウェアベースのクライアントを作成できるかどうかを確認します。

```
# dladm show-linkprop -p rxhwclntavail,txhwclntavail,rxringsavail,txringsavail net4
LINK     PROPERTY        PERM   VALUE   EFFECTIVE  DEFAULT   POSSIBLE
net4     rxhwclntavail   r-     3       3          --        --
net4     txhwclntavail   r-     0       0          --        --
net4     rxringsavail    r-     0       0          --        --
net4     txringsavail    r-     0       0          --        --
```

この出力は、物理データリンク net4 上に 3 つのハードウェアベースの Rx クライアントを作成できることを示しています。

3. VNIC vnic3 (ハードウェアベースの Rx クライアント) を作成します。

```
# dladm create-vnic -l net4 -p rxrings=hw vnic3
```

ハードウェアベースの Tx クライアントの使用可能な数 (txhwclntavail) はゼロであるため、vnic3 の txrings プロパティーを構成することはできません。

4. VNIC vnic3 に割り当てられたリングを確認します。

```
# dladm show-linkprop -p rxrings,txrings vnic3
LINK     PROPERTY       PERM   VALUE     EFFECTIVE  DEFAULT   POSSIBLE
vnic3    rxrings         rw    hw        hw         --        sw,hw
vnic3    txrings         rw    --        8          --        --
```

5. 物理データリンク net4 上に追加のハードウェアベースのクライアントを作成できるかどうかを確認します。

```
# dladm show-linkprop -p rxhwclntavail,txhwclntavail,rxringsavail,txringsavail net5
LINK     PROPERTY        PERM  VALUE     EFFECTIVE   DEFAULT   POSSIBLE
net4     rxhwclntavail   r-    2         2           --        --
net4     txhwclntavail   r-    0         0           --        --
net4     rxringsavail    r-    0         0           --        --
net4     txringsavail    r-    0         0           --        --
```

この出力は、物理データリンク net4 上に 2 つのハードウェアベースの Rx クライアントを作成できることを示しています。

6. 物理データリンク net4 上のリングの使用状況を確認します。

```
# dladm show-phys -H net4
LINK      RINGTYPE    RINGS     CLIENTS
net4      RX          0-3       <default,mcast>
net4      RX          4-7       vnic3
net4      RX          8-11
net4      RX          12-15     --
net4      TX          0-7       <default>,vnic3
```

この出力は、vnic3 が 4 つのリングを排他的に使用できるハードウェアベースの Rx クライアントであることを示しています。Tx リングの場合、vnic3 はリングのデフォルトセットを使用し、さらに物理データリンク net4 上にほかのデータリンクが作成されている場合はそれらとリングを共有します。

プールと CPU の管理

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用に CPU リソースのプールを割り当てることが含まれます。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 プロパティーを構成します。pool リンクプロパティーを使用すると、ネットワークプロセス用に CPU のプールを割り当てることができます。このプロパティーを使用すると、ネットワークリソースの管理を、ゾーン内の CPU 割り当てや管理により適切に統合できます。

リンクの pool プロパティーを設定し、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バインドされます。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、ゾーンに割り当てられていないほかのリンクから使用できなくなります。

注記 - 別のプロパティー cpus 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。cpu および pool プロパティー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ーを設定することはできません。cpus プロパティー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、185 ページの「データリンクに CPU を割り当てる方法」を参照してください。

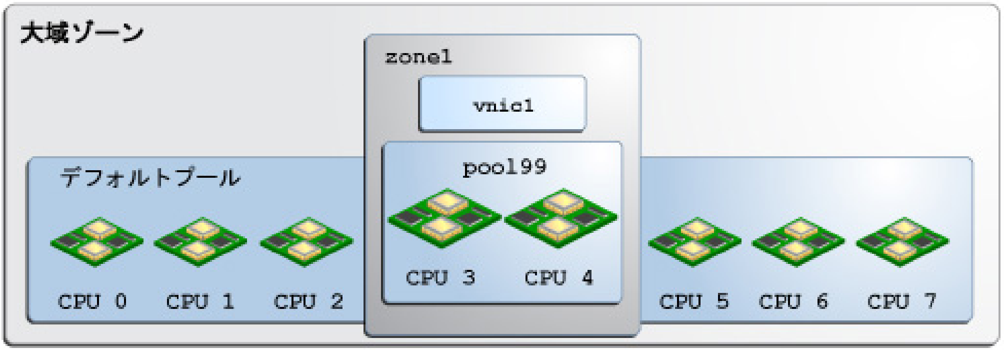
ゾーン内のプールについての詳細は、『Oracle Solaris 11.2 でのリソースの管理 』の第 13 章「リソースプールの作成と管理のタスク」を参照してください。プールの作成および CPU セッ

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。

プールおよび CPU の操作

次の図は、データリンクに pool プロパティー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 7-1 ゾーンに割り当てられた VNIC の pool プロパティー



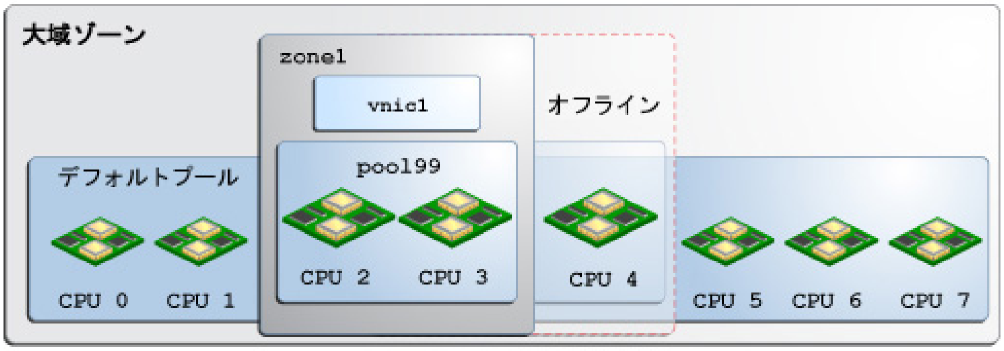
この図では、システムに 8 つの CPU が割り当てられています。システム上にプールが構成されていない場合は、すべての CPU が*デフォルトプール*に属し、大域ゾーンで使用されます。ただし、この例では、CPU 3 と CPU 4 で構成された pool99 プールが作成されています。このプールは、排他的ゾーンである zone1 に関連付けられています。pool99 が vnic1 のプロパティーとして設定された場合、pool99 は vnic1 のネットワークプロセスの管理にも専用で使用されます。vnic1 が zone1 のネットワークインタフェースとして割り当てられたあと、pool99 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 プロパティーは本質的に動的です。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 プロパティー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 プロパティーを設定する必要があります。

たとえば、図7-1「ゾーンに割り当てられた VNIC の pool プロパティー」のシステム CPU 4 がオフラインになったとします。pool プロパティーは動的であるため、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、2 CPU 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。次の図に、更新された構成を示します。

図 7-2 pool プロパティーの自動再構成



dladm show-linkprop コマンドを使用してデータリンクの情報を表示した場合は、pool および cpus データリンクプロパティーの EFFECTIVE 列の値が、それらのプロパティーのシステムで選択された現在の値を示します。

pool および cpus プロパティーに対する、次の読み取り専用の値が表示されます。

- pool データリンクプロパティーの場合、EFFECTIVE 列の値は、ネットワークプロセスに使用されているプールを示します。
- cpus データリンクプロパティーの場合、EFFECTIVE 列の値は、ネットワークプロセスに使用されている CPU を示します。データリンクの cpus プロパティーを表示する方法を示す例については、例7-5「データリンクへの CPU の割り当て」を参照してください。

ゾーンの CPU リソースを管理する場合は、データリンクのプールプロパティーを設定する必要はありません。zonecfg や poolcfg などのコマンドを使用すると、リソースのプールを使用するようにゾーンを構成できます。データリンクに対して cpus および pool リンクプロパティーが設定されていない場合、データリンクの pool および cpus プロパティーの EFFECTIVE 列の値は、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` プロパティーの `EFFECTIVE` 列にはデフォルトプールが表示され、システムは `cpus` プロパティーの `EFFECTIVE` 列の値を選択します。そのため、`dladm show-linkprop` コマンドを使用した場合、`pool` および `cpus` プロパティーの値は空ですが、`pool` および `cpus` プロパティーの `EFFECTIVE` 列には値が表示されます。

また、ゾーンの CPU プールをネットワークプロセス用に割り当てるために、データリンクの `pool` および `cpu` プロパティーを直接設定することもできます。これらのプロパティーを構成したあと、それらの値が `pool` および `cpus` プロパティーの `EFFECTIVE` 列に反映されます。ただし、この代替方法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。

データリンク用の CPU プールの構成

このセクションでは、リンクを作成するとき、またはあとでリンクに追加構成が必要になったときに、データリンクに `pool` プロパティーを設定する方法について説明します。

▼ CPU プールをデータリンク用に構成する方法

始める前に 次のタスクを実行済みである必要があります。

- 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。

注記 - 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 11.2 でのリソースの管理 』の「構成の変更方法」を参照してください。

---

1. **リンクの `pool` プロパティーを、ゾーン用に作成した CPU のプールに設定します。**

   - **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*

   - **VNIC が存在する場合は、次の構文を使用します。**

```
# dladm set-linkprop -p pool=pool VNIC
```

2. **VNIC を使用するようにゾーンを設定します。**

```
global# zonecfg -z zone
zonecfg:zone> add net
zonecfg:zone:net> set physical=VNIC
zonecfg:zone:net> end
```

3. **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:zone> verify
zonecfg:zone> commit
zonecfg:zone> exit
```

例 7-4 ゾーンへのリンクの CPU プールの割り当て

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 7-1「ゾーンに割り当てられた VNIC の pool プロパティー」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に設定されます。

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

# zonecfg -z zone1
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic1
zonecfg:zone1:net> end
zonecfg:zone1> verify
zonecfg:zone1> commit
zonecfg:zone1> exit
```

データリンクへの CPU の割り当て

このセクションでは、cpu プロパティーを構成することによって、データリンクに CPU リソースを割り当てる方法について説明します。リングとは異なり、データリンク用の CPU を排他的に割り当てることはできません。複数のデータリンクに CPU の同じセットを割り当てることができます。

▼ データリンクに CPU を割り当てる方法

1. **管理者になります。**

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **インタフェースに対する CPU 割り当てを確認します。**

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

3. **リンクに CPU を割り当てます。**

データリンクのパケットを処理する CPU のリスト。データリンクの割り込みは、リスト内の CPU のいずれかをターゲットにすることもあります。

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

cpu1,cpu2,... リンクに割り当てる CPU 番号を示します。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. **(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 7-5 データリンクへの 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。

```
# dladm show-linkprop -p cpus net0
LINK     PROPERTY     PERM    VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net0     cpus         rw      --           0-2          --        --
```

この出力は、データリンク `net0` に 3 つの CPU (`0-2`) が暗黙的に割り当てられたことを示します。ただし、CPU はデータリンク `net0` に排他的には割り当てられていません。

```
# dladm set-linkprop -p cpus=0,1 net0
# dladm show-linkprop -p cpus net0
LINK     PROPERTY     PERM    VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net0     cpus         rw      0-1          0-1          --        --
```

この出力は、データリンク `net0` に 2 つの CPU (`0-1`) が明示的に割り当てられたことを示しています。割り当てられた CPU はデータリンク `net0` のパケットを処理します。

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、単一の属性または属性の組み合わせに基づいてネットワークパケットを分類するためのカスタマイズされた方法です。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、23 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。

ネットワークリソースの管理にフローを使用するには、次の手順を実行します。

1. フローの作成。

   フローは、パケットのヘッダー内の情報から導き出される 1 つの属性または属性の組み合わせに基づいて作成されます。

   次の属性のいずれかを使用すると、パケットトラフィックをフローに構成できます。

   - ローカル IP アドレス。
   - リモート IP アドレス。
   - トランスポートプロトコル名 (UDP、TCP、または SCTP)。
   - 差別化サービスフィールド (DS フィールド) 属性 (IPv6 パケットのみで QoS のために使用されます)。詳細は、『Oracle Solaris 11.2 での IP サービス品質の管理 』を参照してください。

   次の属性の組み合わせのいずれかを使用すると、パケットトラフィックをフローに構成できます。

   - トランスポートプロトコル名 (UDP、TCP、または SCTP) とローカル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)。
   - トランスポートプロトコル名 (UDP、TCP、または SCTP) とリモートアプリケーションのポート番号。
   - トランスポートプロトコル名 (UDP、TCP、または SCTP) と、ローカル IP アドレスおよびローカルアプリケーションのポート番号。この属性の組み合わせには、さらに、リモート IP アドレスとリモートアプリケーションのポート番号を含めることができます (`Transport protocol name (UDP, TCP or SCTP) + local IP address + local application port number [+ remote IP address [+ remote application port number]]`)。

   フローはこれらの属性の組み合わせの 1 つだけをベースにすることができます。たとえば、使用されているトランスポートプロトコルとポート (FTP での TCP 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。もっとも一般的なケースとして、トランスポートプロトコル、ローカルまたはリモートの IP アドレス、およびローカルまたはリモートのポートを指定することでフローを作成できます。また、同じリンクに属するすべてのフローに、同じ属性の組み合わせが必要です。

2. ネットワークリソースに関連するプロパティー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。現在は、帯域幅と優先順位のプロパティーをフローに関連付けることができます。

詳細は、188 ページの「フローの構成」を参照してください。

フローへのリソース割り当てのためのコマンド

フローに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、次のとおりです。

- 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、次のコマンド構文を使用します。

  ```
  # flowadm add-flow -l link -a attribute=value[,attribute=value] -p prop=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。パケットトラフィックをフローに構成するために使用できるさまざまな属性のリストについては、186 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。

- 既存のフローのプロパティーを設定するには、次のコマンド構文を使用します。

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

  ここで、*prop* は、フローに割り当てることができるフロープロパティーを示します。フローのプロパティーは、リンクに直接割り当てられているプロパティーと同じです。ただし、フローに関連付けることができるのは、帯域幅と優先順位のプロパティーだけです。これらのプロパティーを構成するには、188 ページの「フローを構成する方法」を参照してください。

詳細は、`flowadm`(1M) のマニュアルページを参照してください。

フローの構成

このセクションでは、フローを作成してフローのプロパティーを設定する方法について説明します。

▼ フローを構成する方法

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **(オプション) 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
```

3. **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
```

4. **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ために使用できる、単一の属性または属性の組み合わせを示します。属性についての詳細は、186 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。

flow 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。

5. **(オプション) 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*

link フローが構成されているデータリンクを表します。

値の範囲は、コマンドの出力の POSSIBLE フィールドの下にリストされます。

6. **適切なフローのプロパティーを設定することによって、フロー上にリソース制御を実装します。**

 # flowadm set-flowprop -p *prop*=*value*[,...] *flow*

リソースを制御するための次のフローのプロパティーを指定できます。

maxbw フローで識別されたパケットが使用できるリンクの帯域幅の最大量。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。

priority 指定されたフローに属するパケットが処理される優先順位。priority プロパティーに対して許可される値は、high、medium、および low です。フローの優先順位が high に設定されている場合は、そのフローに属する

すべてのパケットが、同じリンク上のほかのパケットの前に処理されます。このプロパティーは、待機時間の影響を受けやすいアプリケーションのフローを作成するために使用されます。このプロパティーのデフォルト値は `medium` です。

注記 - 現在のところ、`priority` プロパティーを `medium` から `low` に設定しても、効果はありません。

---

7. **(オプション) データリンク上に作成したフローを表示します。**

```
# flowadm
```

注記 - `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。

---

8. **(オプション) 指定したフローのプロパティー値を表示します。**

```
# flowadm show-flowprop flow
```

このコマンドは、`maxbw` および `priority` のフロープロパティーと、読み取り専用の `hwflow` プロパティーを表示します。

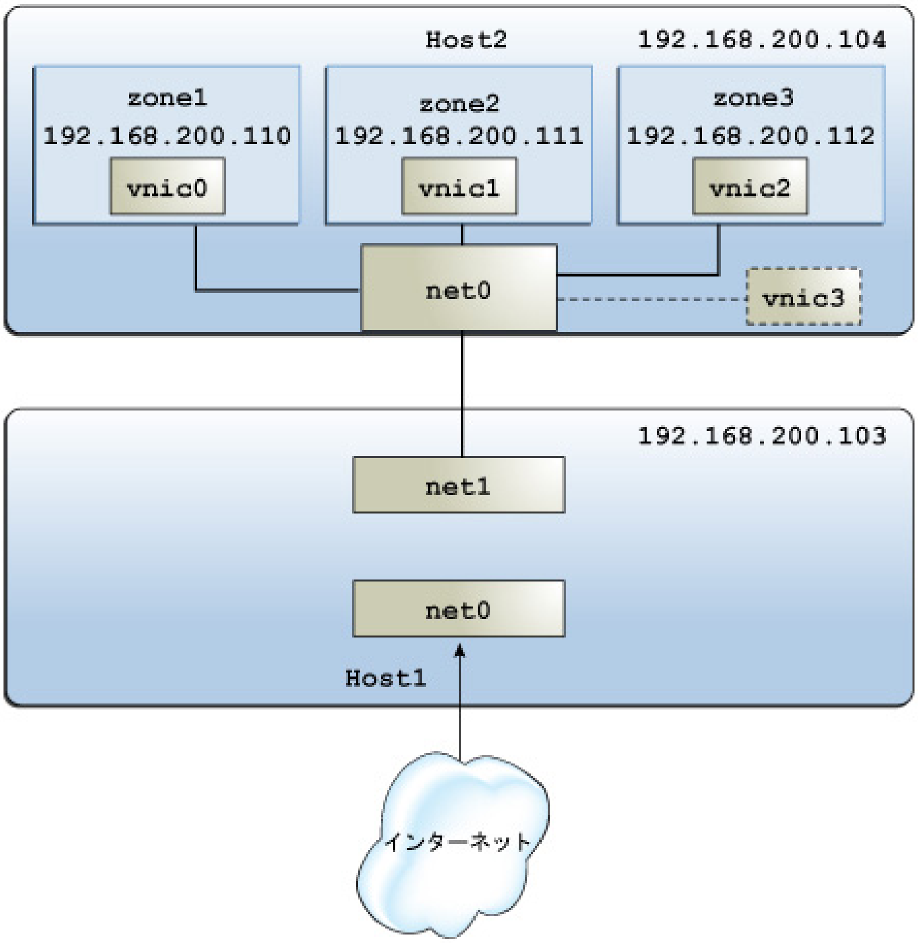
`hwflow` フローのパケット分類を理解するのに役立つ読み取り専用プロパティー。このプロパティーの指定可能な値は、`on` と `off` です。値 `on` は、フローが NIC にオフロードされていて、フローのパケット分類がハードウェアレベルで実行されることを意味します。このプロパティーは、`flowadm add-flow`、`flowadm set-flowprop`、または `flowadm reset-flowprop` コマンドで `-p` オプションとともに使用することはできません。

注記 - 現在、`hwflow` に `on` の値を割り当てることができるのは、トランスポートプロトコル、ローカルまたはリモートの IP アドレス、およびローカルまたはリモートのポートのすべてを指定することよって定義されたフローだけです。また、すべての NIC が `hwflow` プロパティーをサポートするわけではありません。

ユースケース: データリンクとフローのプロパティーを設定することによるネットワークリソースの管理

次のユースケースは、データリンクとフローの両方のプロパティーを設定することによってシステムの効率を向上させるシナリオに基づいています。このユースケースは、次の図に示す構成に基づいています。

図 7-3 データリンクとフローのリソースを管理するためのシステム構成



この図は、互いに接続された次の 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。
  - サーバーおよびルーターとして機能する 1 つの非大域ゾーン。ゾーンに 2 つのインタフェースが割り当てられていて、net0 インタフェースはインターネットに接続し、net1 インタフェースは Host2 を含む内部ネットワークに接続します。

- フローを net1 上に構成して、トラフィックを切り離し、フローに属するパケットがリソースをどのように使用するかの制御を実装します。フローの構成の詳細は、186 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。
- 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。これらの VNIC は、NIC カードが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、171 ページの「NIC リングの管理」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 が、HTTP クライアントとして機能します。残りのゾーン、zone2 と zone3 は、Secure Shell (SSH) プロトコルを介して Host1 へのアクセスを試みる SSH クライアントとして機能します。zone1 のネットワークトラフィックは zone2 および zone3 より多く、時間の影響を受けません。ただし、zone2 および zone3 のネットワークトラフィックは少なく、時間の影響を受けます。したがって、zone2 および zone3 のネットワークトラフィックをより高速に処理するには、zone1 のネットワークトラフィックに割り当てられた帯域幅を制限する必要があります。zone1 に割り当てられた帯域幅を制限しないと、使用可能なすべての帯域幅が使用されます。これにより、残りのゾーン、zone2 および zone3 に対して帯域幅が拒否されることになります。
  - 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、171 ページの「MAC クライアントへのリングの割り当て」を参照してください。

このユースケースのタスクには次のアクションが含まれます。

- フローの作成とフロー制御の構成 – net1 上にフローを作成して、Host1 の net1 によって受信される、フローに属するパケット上に別のリソース制御を作成します。
- Host2 上の VNIC のためのネットワークリソースのプロパティーの構成 – 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。

注記 - このユースケースにはゾーン構成のための手順は含まれていません。ゾーンを構成するには、『Oracle Solaris ゾーンの作成と使用 』の第 1 章「非大域ゾーンの計画および構成方法」を参照してください。

---

1. 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
   lo0/v6       static     ok           --         ::1/128
net1            ip         ok           --         --
   net1/v4      static     ok           --         192.168.200.103/24
net0            ip         ok           --         --
   net0/v4      static     ok           --         10.134.76.129/24
```

2. Host1 の net1 上に次のフローを作成します。
   - httpflow – zone1 と net1 の間のすべての HTTP トラフィックを含めます。

     ```
     # flowadm add-flow -l net1 -a transport=tcp,local_ip=192.168.200.103,\
     local_port=80,remote_ip=192.168.200.110 httpflow
     ```

   - sshflow – net1 との間で送受信されるすべての SSH クライアントを含めます。

     ```
     # flowadm add-flow -l net1 -a transport=tcp,local_ip=192.168.200.103,\
     local_port=22 sshflow
     ```

3. フロー上にリソース制御を実装します。
   - httpflow の場合は、最大帯域幅を 500M に設定します。

     ```
     # flowadm set-flowprop -p maxbw=500M httpflow
     ```

   - sshflow の場合は、優先順位を high に設定します。

     ```
     # flowadm set-flowprop -p priority=high sshflow
     ```

4. 作成したフローに関する情報を確認します。

```
# flowadm
FLOW        LINK     PROTO LADDR            LPORT  RADDR            RPORT DSFLD
httpflow    net1     tcp   192.168.200.103  80     192.168.200.110  --    --
sshflow     net1     tcp   192.168.200.103  22     --               --    --

# flowadm show-flowprop
FLOW         PROPERTY        PERM    VALUE        DEFAULT       POSSIBLE
httpflow     maxbw           rw      500          --            --
httpflow     priority        rw      medium       medium        low,medium,high
httpflow     hwflow          r-      off          --            on,off
```

```
sshflow      maxbw          rw     --          --            --
sshflow      priority       rw     high        medium        low,medium,high
sshflow      hwflow         r-     off         --            on,off
```

出力についての詳細は、flowadm(1M) のマニュアルページを参照してください。

5. Host2 上で、ゾーンごとに net0 上に VNIC を構成します。

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

6. 各 VNIC 上にリソース制御を実装します。

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

7. VNIC をそれぞれのゾーンに割り当てます。

```
# 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
```

8. リングをプライマリインタフェース `net0` と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
```

9. `Host2` の CPU のセットである `pool1` は、`zone1` に割り当てられていると仮定します。`zone1` のネットワークプロセスも管理するために、CPU の同じ `pool1` を割り当てます。

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

◆ ◆ ◆ 第 8 章

ネットワークトラフィックとリソース使用状況のモニタリング

この章では、データリンクとフロー上のネットワークリソースの使用に関するネットワーク統計をモニターするためのタスクについて説明します。ネットワークトラフィックの統計情報をログファイルに記録するように、システム上のネットワークアカウンティングを構成します。この統計情報は、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、ネットワークトラフィックの統計情報を表示するために使用できる 2 つのコマンドについて説明します。

この章の内容は、次のとおりです。

- 197 ページの「データリンクとフローのネットワークトラフィックの統計情報のモニタリングの概要」
- 200 ページの「ネットワークトラフィックの統計情報をモニターするためのコマンド」
- 200 ページの「リンクのネットワークトラフィックの統計情報の表示」
- 206 ページの「フロー上のネットワークトラフィックの統計情報の表示」
- 209 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ティングの構成」

データリンクとフローのネットワークトラフィックの統計情報のモニタリングの概要

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のインバウンド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、アウトバウンドパケットはネットワークに送信されます。

ネットワークトラフィックを管理するために、システムリソースを結合したり、割り当てたりすることができます。データリンクとフローの両方について、受信側および送信側のネットワークトラフィッ

クの統計情報をモニターできます。この章では、主にデータリンクとフロー上の受信側のネットワークトラフィックの統計情報について説明します。

データリンクプロパティーを設定することによって、受信リング、送信リング、およびデータリンク上のその他のリソースを構成できます。データリンク上のネットワークトラフィックによっては、システムでのパケットの処理効率を向上させるために、データリンクに専用ハードウェアリングを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データリンクにより多くのリングを割り当てることができます。データリンクにハードウェアリングを割り当てる方法の詳細は、175 ページの「クライアントの構成とリングの割り当て」を参照してください。

次の理由で、データリンクに専用ハードウェアリングを割り当てられないことがあります。

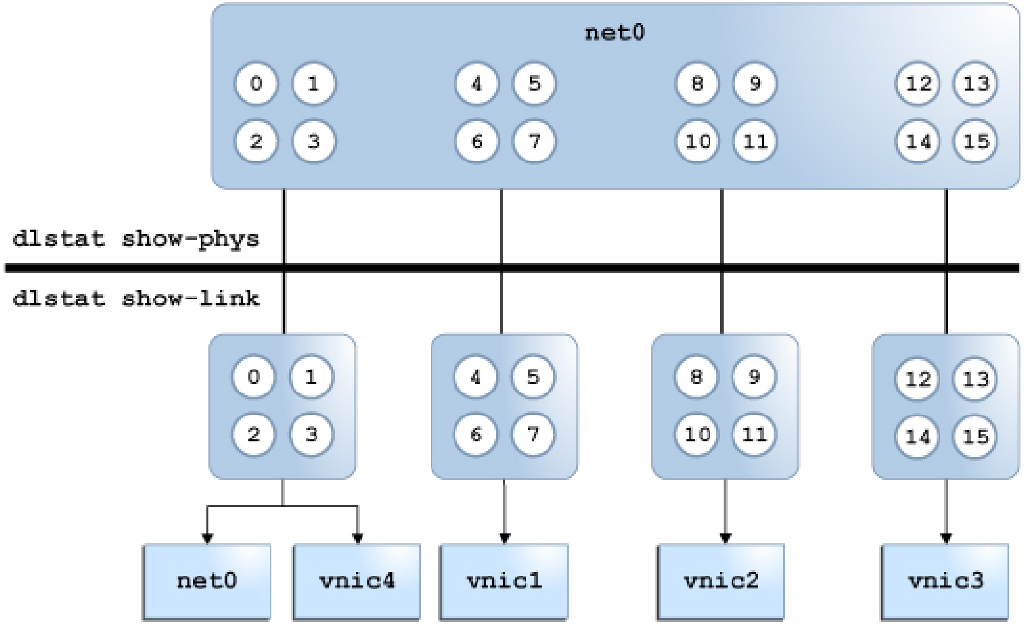
- ハードウェアリソースの不足。たとえば、データリンクに排他的に割り当てることができる使用可能なリングが存在しない可能性があります。
- ハードウェア機能の不足。たとえば、NIC がハードウェアリングを公開していない可能性があります。
- データリンクが低レベルのハードウェアデータリンクに結合されていない可能性があります。たとえば、etherstub 上に VNIC を作成する場合が該当します。

一部のデータリンクは、次の理由で、リングを共有するように構成されることがあります。

- 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る。
- NIC でリング割り当てがサポートされていない可能性がある。
- データリンクがリング割り当てをサポートしているにもかかわらず、排他的使用のためにリングを割り当てることができなくなった。

次の図は、データリンク間でのハードウェアリングの割り当てを示しています。

図 8-1　　データリンクでのリングの割り当て



この図は、次の構成を示しています。

- net0 データリンクには、ほかのデータリンクに割り当てることができる 16 個のハードウェアリング (0-15) が存在します。
- vnic1、vnic2、vnic3、vnic4 の各 VNIC は、データリンク net0 上に構成されています。
- vnic1、vnic2、vnic3 の各 VNIC には、それぞれ 4 つの専用ハードウェアリングが割り当てられています。
- ハードウェアリング (0-3) は、データリンク net0 と VNIC vnic4 の間で共有されています。

  次の例は、物理データリンク net0 へのリング割り当てを示しています。

```
# dladm show-phys -H net0
LINK        RINGTYPE  RINGS          CLIENTS
net0        RX        0-3            <default,mcast>,vnic4
net0        RX        4-7            vnic1
net0        RX        8-11           vnic2
net0        RX        12-15          vnic3
```

```
net0        TX       0-7                 <default>,vnic4,vnic3,vnic2,vnic1
```

- 物理データリンク net0 のネットワークトラフィックの統計情報を表示するには、dlstat show-phys コマンドを使用します。例8-1「システム上の物理リンクのトラフィック統計情報の表示」を参照してください。
- データリンク net0、vnic1、vnic2、vnic3、および vnic4 のネットワークトラフィックの統計情報を表示するには、dlstat show-link コマンドを使用します。例8-7「専用ハードウェアリングが割り当てられているデータリンクのネットワークトラフィックの統計情報の表示」を参照してください。

ネットワークトラフィックの統計情報をモニターするためのコマンド

dlstat コマンドと flowstat コマンドを使用すると、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の統計情報をモニターできます。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、管理コマンドのペアとモニタリングコマンドのペアの機能比較を示します。

管理コマンド		モニタリングコマンド	
コマンド	機能	コマンド	機能
dladm	データリンクを構成および管理します	dlstat	データリンク上のトラフィックの統計情報を表示します
flowadm	フローを構成および管理します	flowstat	フロー上のトラフィックの統計情報を表示します

リンクのネットワークトラフィックの統計情報の表示

dlstat コマンドの次のバリアントを使用すると、ネットワークトラフィック情報を表示できます。

コマンド	提供される情報
dlstat [*link*] dlstat -rt [*link*]	データリンク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統計情報を表示します

コマンド	提供される情報
`dlstat show-link` [*link*]	
`dlstat show-link -rt` [*link*]	リングごと、データリンク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統計情報を表示します
`dlstat show-phys` [*link*]	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を表示します
`dlstat show-phys -rt` [*link*]	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を表示します
`dlstat show-aggr` [*link*] `dlstat show-aggr -rt` [*link*]	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を表示します
`dlstat show-bridge` [*bridge*] `dlstat show-bridge -rt` [*bridge*]	ブリッジ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を表示します

`-r` オプションを使用すると受信側の統計情報を表示でき、`dlstat` コマンドで `-t` オプションを使用すると送信側の統計情報を表示できます。その他のオプションについての詳細は、`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のネットワークトラフィックの統計情報の表示

`dlstat show-phys` 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8-1「データリンクでのリング割り当て」に示すように、`dlstat show-phys` 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。

次のコマンド構文を使用すると、ネットワークデバイス上のネットワークトラフィック統計情報を表示できます。

```
# dlstat show-phys [-r|-t] [-Tu | -Td] [link] [interval [count]]
```

`-r`
: 受信側のネットワークトラフィックの統計情報のみを表示します。`-t` オプションをこのオプションと一緒に指定しないでください。

  `-r` オプションまたは `-t` オプションを指定しない場合は、送信側と受信側の両方のネットワークの統計情報が表示されます。

`-t`
: 送信側のネットワークトラフィックの統計情報のみを表示します。`-r` オプションをこのオプションと一緒に指定しないでください。

	-r または -t オプションを指定しない場合は、送信側と受信側の両方のネットワークの統計情報が表示されます。
-Tu	現在の時間を内部表現で表示します。
-Td	現在の時間を標準の日付形式で表示します。
link	ネットワーク統計をモニターするデータリンクの名前。データリンクを指定しない場合は、システムに構成されているすべてのデータリンクに関する情報が表示されます。
interval	ネットワーク統計をリフレッシュする時間を秒数で指定します。
count	表示されているネットワークトラフィックの統計情報がリフレッシュされる回数を指定します。カウント値を指定しない場合、統計情報は無期限にリフレッシュされます。

例 8-1 システム上の物理リンクのトラフィック統計情報の表示

この例では、システムの各リンク上の受信と送信の両方のネットワークトラフィックが表示されます。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。

```
# dlstat show-phys
LINK    IPKTS   RBYTES     OPKTS    OBYTES
net5        0        0         0         0
net6        0        0         0         0
net0   25.57K    5.10M     1.93K   226.05K
net0      179   26.63K       161    22.75K
net3        0        0         0         0
net4        0        0         0         0
net2        0        0         0         0
net8      238  137.16K       191     8.41K
net1        0        0         0         0
...
```

出力には次の情報が表示されます。

LINK	物理および仮想データリンク (名前で識別されます)
IPKTS	リンク上のインバウンドパケットの数
RBYTES	リンク上で受信されたバイト数
OPKTS	リンク上のアウトバウンドパケットの数
OBYTES	このリンク上で送信されたバイト数

例 8-2 ネットワークデバイスの受信側のトラフィック統計情報の表示

この例では、受信されるネットワークトラックの統計情報が、間隔値 2 秒、カウント値 3 で表示されます。

```
# dlstat show-phys -r 2 3
LINK   TYPE  INDEX    IPKTS   RBYTES
net0     rx      0    8.03M   12.09G
net1     rx      0        0        0
net0     rx      0    8.79K   13.28M
net1     rx      0        0        0
net0     rx      0    8.50K   12.83M
net1     rx      0        0        0
```

データリンク net0 および net1 は 1 つのセットと考えてください。最初のデータリンク net0 および net1 のセットは、受信したパケットとバイトの合計数を示しています。この例では、8.03M が受信したパケットの合計数で、12.09G が net0 で受信したバイトの合計数です。2 番目のデータリンク net0 および net1 のセットは、正規化された値とも呼ばれる、ネットワークトラフィックの統計情報の 1 秒あたりの変化量を示しています。つまり、8.79K は、net0 によって 2 秒間隔で受信されたパケットの正規化された値です。同様に、3 番目のデータリンク net0 および net1 のセットも、2 秒間隔のネットワークトラフィックの統計情報の正規化された値を示しています。

例 8-3 ネットワークデバイスの受信側のトラフィック統計情報の表示

この例では、データリンク net0 の受信トラフィックの統計情報が表示されます。

```
# 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
```

この例では、net0 データリンクに 8 つの受信リングがあり、それらが INDEX フィールドで特定されています。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、各リンク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。詳細は、171 ページの「NIC リングの管理」を参照してください。

例 8-4 ネットワークデバイスの送信側のトラフィック統計情報の表示

この例では、ネットワークデバイスとしての net0 の送信リングの使用状況が表示されます。

```
# dlstat show-phys -t net0
LINK  TYPE  INDEX    OPKTS   OBYTES
net0    tx      0       93    4.63K
net0    tx      1        0        0
net0    tx      2        0        0
net0    tx      3        0        0
net0    tx      4        0        0
net0    tx      5       47   11.02K
net0    tx      6       23    7.13K
net0    tx      7        0        0
```

例 8-5 時間を含むネットワークデバイスのトラフィックの統計情報の表示

次の例では、内部表現の現在の時間とともに、ネットワークデバイスとしての net0 のネットワークトラフィックに関する統計情報が表示されます。

```
# dlstat show-phys -Tu net0
1401652481
           LINK    IPKTS   RBYTES    OPKTS   OBYTES
           net0      184   27.14K      165   22.91K
```

次の例では、標準の日付形式の現在の時間とともに、ネットワークデバイスとしての net0 のネットワークトラフィックに関する統計情報が表示されます。

```
# dlstat show-phys -Td net0
Sun Jun  1 12:54:47 PDT 2014
           LINK    IPKTS   RBYTES    OPKTS   OBYTES
           net0      184   27.14K      165   22.91K
```

データリンクのネットワークトラフィックの統計情報の表示

dlstat show-link コマンドを使用すると、データリンクのネットワークトラフィックの統計情報を表示できます。

例 8-6 データリンクのネットワークトラフィックの統計情報の表示

この例は、データリンク vnic0 のネットワークトラフィックの統計情報を示しています。

```
# dlstat show-link vnic0
LINK     IPKTS   RBYTES    OPKTS   OBYTES
vnic0    3       180       0       0
```

例 8-7 専用ハードウェアリングが割り当てられているデータリンクのネットワークトラフィックの統計情報の表示

この例は、4 つの専用 Rx リングが割り当てられているデータリンク vnic0 の受信側のネットワークトラフィックの統計情報を示しています。出力の ID 列にある hw 値は、データリンク vnic0 に専用ハードウェアリングが割り当てられていることを示します。

```
# dlstat show-link -r vnic0
LINK   TYPE      ID  INDEX    IPKTS   RBYTES    INTRS    POLLS   IDROPS
vnic0    rx   local     --        0        0        0        0        0
vnic0    rx   other     --       64    2.94K        0        0        0
vnic0    rx      hw      8        0        0        0        0        0
vnic0    rx      hw      9       53    7.97K       53        0        0
vnic0    rx      hw     10        4      392        4        0        0
vnic0    rx      hw     11  153.65K  220.68M  153.65K        0        0
```

例 8-8 データリンクの送信側のネットワークトラフィックの統計情報の表示

この例は、データリンク vnic0 の送信側のネットワークトラフィックの統計情報を示しています。

```
# dlstat show-link -t vnic0
LINK   TYPE      ID  INDEX    OPKTS   OBYTES   ODROPS
vnic0    tx   local     --        0        0        0
vnic0    tx   other     --       19      798        0
vnic0    tx      sw     --        0        0        0
```

例 8-9 専用ハードウェアリングが割り当てられていないデータリンクのネットワークトラフィックの統計情報の表示

この例は、専用 Rx リングが割り当てられていないデータリンク net6 のネットワークトラフィックの統計情報を示しています。出力の ID 列にある sw 値は、データリンク net6 が専用ハードウェアリングを使用して構成されていないことを示します。

```
# dlstat show-link -r net6
LINK  TYPE      ID  INDEX    IPKTS   RBYTES INTRS POLLS   IDROPS
net6    rx   local     --        0   0        0     0        0
net6    rx   other     --        0   0        0     0        0
net6    rx      sw     --        0   0        0     0        0
```

リンクアグリゲーションのネットワークトラフィックの統計情報の表示

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。

例 8-10 リンクアグリゲーションのネットワークトラフィックの統計情報の表示

```
# dlstat show-aggr
LINK       PORT     IPKTS    RBYTES     OPKTS    OBYTES
aggr0      --          13       832        13       780
aggr0      net0         0         0        13       780
aggr0      net3        13       832         0         0
```

この例の出力は、ベースとなる 2 つのリンク net0 と net3 を含むリンクアグリゲーション aggr0 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。

リンクアグリゲーションの詳細については、『Oracle Solaris 11.2 でのネットワークデータリンクの管理 』の第 2 章「リンクアグリゲーションを使用した高可用性の構成」を参照してください。

ブリッジのネットワークトラフィックの統計情報の表示

dlstat show-bridge コマンドは、各ブリッジのネットワーク統計情報を表示し、各ブリッジに接続されたリンクの統計情報の一覧を表示します。

例 8-11 ブリッジのネットワークトラフィックの統計情報の表示

この例では、ブリッジ rbblue0 および stbred0 のネットワーク統計情報が表示されます。

```
# dlstat show-bridge
BRIDGE         LINK      IPKTS    RBYTES     OPKTS    OBYTES     DROPS  FORWARDS
rbblue0          --      1.93K   587.29K     2.47K     3.30M         0         0
              simblue1      72     4.32K     2.12K     2.83M         0        --
              simblue2   1.86K   582.97K       348   474.04K         0        --
stbred0          --        975   976.69K     3.44K     1.13M         0        38
               simred3     347   472.54K     1.86K   583.03K         0        --
               simred4     628   504.15K     1.58K   551.51K         0        --
```

フロー上のネットワークトラフィックの統計情報の表示

フロー上の統計情報は、システムに定義されているすべてのフロー上のパケットトラフィックを評価する際に役立ちます。フロー上の統計情報を表示するには、flowstat コマンドを使用します。詳細は、flowstat(1M) のマニュアルページを参照してください。

フロー上のネットワークトラフィックの統計情報を表示するには、次のコマンド構文を使用します。

flowstat [-r|-t] [-l *link*] [-Tu | -Td] [*flow*] [*interval* [*count*]]

`-r`
: 受信側のネットワークトラフィックの統計情報のみを表示します。-t オプションをこのオプションと一緒に指定しないでください。

  -r オプションまたは -t オプションを指定しない場合は、送信側と受信側の両方のネットワークの統計情報が表示されます。

`-t`
: 送信側のネットワークトラフィックの統計情報のみを表示します。-r オプションをこのオプションと一緒に指定しないでください。

  -r オプションまたは -t オプションを指定しない場合は、送信側と受信側の両方のネットワークの統計情報が表示されます。

`-l` *link*
: ネットワーク統計をモニターするデータリンクの名前。データリンクを指定しない場合は、システムに構成されているすべてのフローに関する情報が表示されます。

`-Tu`
: 現在の時間を内部表現で表示します。

`-Td`
: 現在の時間を標準の日付形式で表示します。

flow
: ネットワークの統計情報をモニターするフローの名前。フローを指定しない場合は、指定したリンクに応じて、すべてのフローの統計情報が表示されます。

interval
: ネットワーク統計をリフレッシュする時間を秒数で指定します。間隔値を指定しない場合は、パケットとバイトの合計数も表示されます。

count
: 表示されているネットワークトラフィックの統計情報がリフレッシュされる回数を指定します。カウント値を指定しない場合、統計情報は無期限にリフレッシュされます。

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 8-12 フローのネットワークトラフィックの統計情報の表示

この例では、システムに構成されているすべてのフローのネットワークトラフィック統計情報が、間隔値 1 秒、カウント値 2 で表示されます。

```
# flowstat 1 2
FLOW      IPKTS   RBYTES   IDROPS    OPKTS   OBYTES   ODROPS
flow1     1.78M    2.68G      443  889.57K   58.72M        0
flow2         0        0        0        0        0        0
flow1     8.31K   12.51M      243    4.22K  280.45K        0
```

```
flow2        0        0        0        0        0        0
```

フロー flow1 および flow2 は 1 つのセットと考えてください。最初のフロー flow1 および flow2 のセットは、フローによって送受信されたネットワークトラフィックの統計情報の合計数を示しています。この例では、1.78M は flow1 によって受信されたパケットの合計数です。2 番目のフロー flow1 および flow2 のセットは、正規化された値とも呼ばれる、ネットワークの統計情報の 1 秒あたりの変化量を示しています。この例では、8.31K は、flow1 によって 1 秒間隔で受信されたパケットの正規化された値です。

例 8-13 フローの送信側トラフィックの統計情報の表示

この例では、システムに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ネットワークトラフィック統計情報が表示されます。

```
# flowstat -t
FLOW     OPKTS   OBYTES   ODROPS
flow1   24.37M    1.61G        0
flow2        0        0        0
```

例 8-14 データリンク上のフローの受信側トラフィックの統計情報の表示

この例では、データリンク net0 に構成されているすべてのフローの受信ネットワークトラフィックが、間隔値 2 秒、カウント値 5 で表示されます。

```
# flowstat -r -l net0 2 5
FLOW     IPKTS   RBYTES   IDROPS
flow1    2.38M    3.59G   14.89K
flow2        0        0        0
flow1    8.24K   12.40M      180
flow2        0        0        0
flow1    8.94K   13.47M      206
flow2        0        0        0
flow1    7.43K   11.19M      161
flow2        0        0        0
flow1    8.38K   12.62M      213
flow2        0        0        0
```

フロー flow1 および flow2 は 1 つのセットと考えてください。最初のフロー flow1 および flow2 のセットは、フローによって受信されたパケットとバイトの合計数を示しています。この例では、2.38M が受信されたパケットの合計数で、3.59G が flow1 で受信されたバイトの合計数です。2 番目のフロー flow1 および flow2 のセットは、正規化された値とも呼ばれる、ネットワークの統計情報の 1 秒あたりの変化量を示しています。この例では、8.24K は、flow1 によって 2 秒間隔で受信されたパケットの正規化された値です。同様に、そのあとのフローのセットは、2 秒の定期的な間隔で、ネットワークトラフィックの統計情報の正規化された値を示しています。

例 8-15 時間を含むフローのトラフィックの統計情報の表示

次の例では、内部表現の現在の時間とともに、データリンク `net0` 上に作成されたすべてのフロー上の受信トラフィックに関する統計情報が表示されます。

```
# flowstat -r -l net0 -Tu
1364380279
          FLOW        IPKTS    RBYTES    IDROPS
          tcp-flow  183.11K  270.24M         0
          udp-flow        0        0         0
```

次の例では、標準の日付形式の現在の時間とともに、データリンク `net0` 上に作成されたすべてのフロー上の受信トラフィックに関する統計情報が表示されます。

```
# flowstat -r -l net0 -Td
Wednesday, March 27, 2013 04:01:011 PM IST
          FLOW        IPKTS    RBYTES    IDROPS
          tcp-flow  183.11K  270.24M         0
          udp-flow        0        0         0
```

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。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。詳細は、acctadm(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。

▼ ネットワークアカウンティングを設定する方法

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」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、次のアカウンティングタイプに対応しています。

- `process` – プロセスアカウンティング
- `task` – タスクアカウンティング
- フロー – フローアカウンティング
- `net` – ネットワークアカウンティング

注記 - ネットワークアカウンティングはまた、186 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」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 8-16 システム上でのネットワークアカウンティングの設定

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ティングを構成する方法を示しています。

次のように、すべてのアカウン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
              Net 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ティングがアクティブになっていないことを示します。したがって、拡張ネットワークアカウンティングを有効にする必要があります。

```
# 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の表示

ネットワークアカウンティングを有効にしたら、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。

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ティング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186 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」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。

データリンク上のネットワークトラフィックの履歴統計情報の表示

次のコマンド構文を使用すると、データリンクのネットワークトラフィックの履歴統計情報を表示できます。

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 *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*　ネットワーク統計情報のログに記録された情報を表示する開始時間を指定します。MM/DD/YYY,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。

-e *end-time*　ネットワーク統計情報のログに記録された情報を表示する終了時間を指定します。MM/DD/YYY,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。

例 8-17　データリンク上のリソース使用状況に関する履歴統計情報の表示

この例では、システムのすべてのデータリンク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が表示されます。

```
# dlstat show-link -h -f /var/log/net.log
LINK  DURATION  IPKTS    RBYTES     OPKTS  OBYTES     BANDWIDTH
net0   80        1031    546908      0        0       2.44 Mbps
net1   100       2045    235977      0        0       9.67 Mbps
```

フロー上のネットワークトラフィックの履歴統計情報の表示

次のコマンド構文を使用すると、フローのネットワークトラフィックの履歴統計情報を表示できます。

```
# 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*	ネットワーク統計情報のログに記録された情報を表示する開始時間を指定します。MM/DD/YYY,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。
-e *end-time*	ネットワーク統計情報のログに記録された情報を表示する終了時間を指定します。MM/DD/YYY,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 8-18 フロー上のリソース使用状況に関する履歴統計情報の表示

次の例では、リソース使用状況の履歴統計情報をシステムの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 flowtcp 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、gnuplot 形式を使用して、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 flowtcp 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 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

あ

アクセス制御リスト (ACL), 80
アップリンクポート
　定義, 102
移行
　VF VNIC, 56
　VNIC, 48
インストール
　エッジ仮想ブリッジング, 84
エッジ仮想ブリッジング, 80 *参照* EVB
　Ethernet リンクの VDP 状態, 92
　EVB によるネットワークおよびサーバーの効率の向上の仕組み, 81
　EVB を使用する場合と使用しない場合のサーバー効率, 83
　Oracle VSI マネージャー, 90
　`oracle_v1`, 90
　`oracle_v1` エンコーディング, 90
　VDP 状態の表示例, 92
　VDP 統計情報の表示例, 93
　VDP の使用, 90
　VDP プロトコル, 90
　VM 間の通信を管理するための LLDP の使用, 88
　VM での通信の制御, 85
　VNIC 情報交換の仕組み, 91
　VNIC 情報の交換, 90
　VSI 識別子, 90
　VSI タイプ ID, 90, 94
　VSI バージョン, 90, 94
　VSI プロファイル, 90
　VSI マネージャー, 90
　VSI マネージャー ID, 90
　アクセス制御リスト, 80
　インストール, 84
　外部スイッチ経由での VM の通信の有効化, 85
　概要, 80
　自動仮想ポート構成, 81
　デフォルトの EVB 構成の変更, 94
　データリンクプロパティー, 94
　反射型リレー, 80, 80
　リンクプロパティーの表示例, 93
エッジ交換ブリッジング
　コンポーネント, 90
エラスティック仮想スイッチ, 100, 115
　IP アドレスのブロックの関連付け, 102
　VLAN に基づいた構成, 154
　VNIC `anet` リソースの作成, 138
　VNIC の作成, 137
　概要, 99
　仮想ポートの関連付け, 102
　管理, 139
　計画, 120
　削除, 150
　作成, 135
　テナント用の VXLAN に基づいた構成, 161
　必須パッケージ, 133
　表示, 135, 140
　プロパティーの設定, 141
　プロパティーの表示, 142
　モニタリング, 151
　リソース
　　IP ネットワーク, 104
　　仮想ポート, 104

か

外部仮想ネットワーク, 14
拡張アカウンティング機能, 209
　タスクアカウンティング, 210
　プロセスアカウンティング, 210
　フローアカウンティング, 210

リンクとフローのネットワークアカウンティング, 210
仮想拡張ローカルエリアネットワーク *参照* VXLAN
仮想スイッチ, 15, 15, 100
仮想ネットワーク, 14, 18, 30, 60
参照 VNIC
etherstub, 15
etherstub の構成, 27
VNIC, 15
VNIC の構成, 27
外部および内部, 14
仮想スイッチ, 15
構築, 30
コンポーネント, 15
実装, 21
ゾーン, 15, 30
ゾーンの構成, 31
ゾーンの再構成, 33
排他的な IP タイプのゾーン, 35
プライベート, 14
仮想ネットワークインタフェースカード *参照* VNIC
仮想ネットワークのコンポーネント
構成, 25
仮想ポート, 100 *参照* VPort
仮想マシン, 100
仮想リンク状態
表示, 43
仮想 NIC *参照* VNIC
管理
EVS クライアントプロパティー, `evsadm` サブコマンド, 113
EVS コントローラプロパティー, `evsadm` サブコマンド, 113
IPnet, `evsadm` サブコマンド, 112
VPort, `evsadm` サブコマンド, 112
エラスティック仮想スイッチ, `evsadm` サブコマンド, 111
擬似 Ethernet NIC *参照* etherstub
構成
VXLAN, 67
コマンド
EVS の管理, 111
仮想ネットワークのコンポーネントの構成, 25
データリンクへのリソースの割り当て, 170
ネットワークトラフィックの統計情報のモニタリング, 200
フローへのリソース割り当て, 188
リングの管理, 172

さ

サービスの品質 (QoS), 21, 169
サービスレベル契約 *参照* SLA
作成
IP インタフェース, 28
サブネット, 104
システムで作成された VNIC, 26
表示, 42
自動的に生成される VXLAN データリンク, 115
受信リング *参照* Rx リング
シングルルート I/O 仮想化 *参照* SR-IOV
送信リング *参照* Tx リング
ゾーン, 15, 110
VNIC の一時的な作成, 36
VNIC の割り当て, 34
VXLAN の使用, 64
VXLAN の割り当て, 72
仮想ネットワークのための再構成, 33
ネットワーク仮想化のための構成, 31
ゾーンでの VNIC、VLAN、および IPoIB パーティションの一時的な作成, 37
ゾーンでの VNIC の一時的な作成, 37
ゾーンに接続された VNIC の削除, 52
ソフトウェアベースのクライアント, 171, 171

た

帯域幅
データリンク上での設定, 22
フロー上での設定, 22
フロー上の設定, 189
データセンターの統合, 21
データリンク
CPU の割り当て, 22
CPU プールの割り当て, 22
EVB プロパティー, 94
帯域幅の割り当て, 22
ネットワークトラフィックの履歴統計情報の表示, 211
リソース制御プロパティー, 22, 170
データリンクプロパティー
`cpu`, 169

maxbw, 169
pool, 169
rxrings と txrings, 169
データリンクへの CPU の割り当て, 185
テナント
定義, 106

な

内部仮想ネットワーク, 14
ネットワークアカウンティング, 209, 210
ネットワークアカウンティングの構成, 209
ネットワーク仮想化, 13
etherstub, 16
仮想スイッチ, 15
仮想ネットワーク, 13
実装のための dladm サブコマンド, 170
ゾーン, 17
データセンターの統合, 21
ネットワークリソースの管理, 13
ネットワークデータリンクの管理
EVB, 21
ネットワークトラフィックの統計情報
データリンクの表示, 204
ネットワークデバイスの表示, 201
ブリッジの表示, 206
フローの表示, 206
リンクアグリゲーションの表示, 205
リンクの表示, 200
ネットワークトラフィックの履歴統計情報の表示
データリンク, 211
フロー, 211
ネットワークの統計情報, 200
ネットワーク使用のモニタリング, 197
履歴トラフィック情報, 210
ネットワークリソース
CPU, 21
CPU プール, 21
NIC リング, 21
帯域幅, 21
優先順位, 21
ネットワークリソースの管理, 13, 21, 169, 169
CPU, 181
CPU プール, 181
NIC リング, 171
実装のための dladm サブコマンド, 170
データリンクとフローのプロパティーを設定すること
によって, 190
データリンクプロパティーの使用, 169
データリンクプロパティーを使用した, 21, 22
フローの使用, 186
フローを使用した, 21, 23
利点, 23
ネットワークレーン, 21

は

ハードウェアベースのクライアント, 171
ハードウェアリング, 171
反射型リレー, 16, 80
表示
EVB 関連のデータリンクプロパティー, 97
EVS コントローラプロパティー, 130
IPnet, 144
VDP と ECP の状態と統計情報, 92
VPort, 148
エラスティック仮想スイッチの情報, 132, 135,
140
仮想リンク状態, 43
物理リンク状態, 43
物理リンク状態
表示, 43
物理リンクでの EVB 関連のデータリンクプロパティー
の表示例, 97
プライベート仮想ネットワーク, 14, 16
etherstub を使用した構成, 37
プライマリクライアント, 171, 174
フロー, 23, 26, 186
構成, 188
作成, 188, 189
情報の表示, 190
属性に基づいた, 187
帯域幅の設定, 188, 189
ネットワークトラフィックの履歴統計情報の表示,
212
プロパティーの設定, 188
優先順位の設定, 188
フロー制御, 186
フローの構成, 188
コマンド, 188
フローを構成するための属性
DS フィールド, 187

- トランスポートプロトコル, 187
- トランスポートプロトコルとアプリケーションのポート番号, 187
- トランスポートプロトコルと、アプリケーションのポート番号および IP アドレス, 187
- リモート IP アドレス, 187
- ローカル IP アドレス, 187

プロトコル
- DCBX, 81
- ECP, 90
- VDP, 90

変更
- デフォルトの EVB 構成, 94

ま
モニタリング
- ネットワーク使用, 197
- ネットワークトラフィックの統計情報
  - コマンド, 200

や
有効化
- 外部スイッチ経由での VM の通信, 85

優先順位
- データリンク, 22
- フロー, 22, 188

ら
リモートアプリケーションのポート番号, 187
リモート IP アドレス, 187
リングの管理
- コマンド, 172

リング割り当て, 171, 171
- VLAN, 171
- リングの使用とリング割り当て, 174

レイヤー 2 ネットワーク, 60
ローカルアプリケーションのポート番号, 187
ローカル IP アドレス, 187

A
`acctadm` コマンド, 209

C
CPU, 22, 181
- CPU 割り当て, 185

CPU の管理, 181
CPU プール, 181
- 操作, 182
- デフォルトプール, 182
- データリンク, 22
- リンクへの割り当て, 184

D
DCBX プロトコル, 81
`dladm` コマンド, 110, 114
- `create-etherstub`, 27, 38
- `create-vlan`, 171
- `create-vnic`, 25, 114, 170, 184
- `create-vxlan`, 67
- `delete-vnic`, 50
- `delete-vxlan`, 72
- `modify-vnic`, 45, 46, 49
- `set-linkprop`, 29, 170
- `show-ether`, 92
- `show-link`, 72
- `show-linkprop`, 53, 172, 174
- `show-phys`, 57, 173, 174
- `show-vnic`, 41, 42, 57, 114
- `show-vxlan`, 67, 72

`dlstat show-ether` コマンド, 93
`dlstat` コマンド, 26, 200, 200
- `show-aggr`, 205
- `show-bridge`, 206
- show-link, 211
- `show-phys`, 201

DS フィールド, 187

E
ECP の状態と統計情報の表示例, 94
Edge Control Protocol (ECP), 90
- ECP の状態と統計情報の表示, 94

etherstub, 15, 16
- 構成, 27
- 作成, 27

プライベート仮想ネットワーク, 37
EVB, 21 *参照* エッジ仮想ブリッジング
EVB 関連のデータリンクプロパティーの設定例, 96
EVS
管理タスク, 119
セキュリティー要件, 117
ゾーンでの機能, 117
名前空間の管理, 106
必須パッケージ, 116
利点, 103
EVS クライアント, 110
EVS コントローラ, 108
計画, 121
構成, 125
作成と管理, 121
設定, 130
必須パッケージ, 116, 122
表示, 130
プロパティーの設定, 130
プロパティーの表示, 130
EVS コントローラのプロパティー, 108
EVS コントローラプロパティー
設定, 125
表示, 125
EVS ノード
定義, 111
EVS のコンポーネント, 106
EVS クライアント, 106
EVS コントローラ, 106
EVS ノード, 106
EVS マネージャー, 106
EVS マネージャー
説明, 107
`evsadm`
EVS クライアントプロパティーを管理するためのサブコマンド, 113
EVS コントローラプロパティーを管理するためのサブコマンド, 113
IPnet を管理するためのサブコマンド, 112
VPort を管理するためのサブコマンド, 112
エラスティック仮想スイッチを管理するためのサブコマンド, 111
`evsadm` コマンド, 110, 111
`add-ipnet`, 112, 135
`add-vport`, 112, 135
`create-evs`, 111, 135
`delete-evs`, 111, 150
`remove-ipnet`, 112, 144
`remove-vport`, 112, 150
`reset-vport`, 113, 149
`set-controlprop`, 113, 130
`set-evsprop`, 112, 112, 141
`set-prop`, 107, 108, 113, 130
`set-vportprop`, 146
`show-controlprop`, 113, 130
`show-evs`, 112, 140
`show-evsprop`, 112, 142
`show-ipnet`, 112, 144
`show-prop`, 113, 130
`show-vport`, 112, 148
`show-vportprop`, 112, 146
`evsstat` コマンド, 151
エラスティック仮想スイッチのネットワークトラフィック統計情報の表示, 113

F
`flowadm` コマンド, 186, 190
`add-flow`, 188, 189, 193
`help`, 188
`set-flowprop`, 188, 189, 193
`show-flowprop`, 190, 193
`flowstat` コマンド, 26, 200, 206

G
GARP VLAN 登録プロトコル (GVRP), 29

I
IP アドレス, 104
IP ネットワーク *参照* IPnet
`ipadm` コマンド, 192
`create-addr`, 28
`create-ip`, 28
IP 情報の表示, 192
`show-addr`, 67
IPnet
エラスティック仮想スイッチへの追加, 135
管理, 139, 144

削除, 144
表示, 144
IPoIB
ゾーンでの一時的な作成, 37

L

LLDP, 81
VM 間の通信を管理するための LLDP の使用, 88

M

MAC アドレス、VNIC 上の, 15
MAC クライアント, 171
構成, 175
ソフトウェアベース, 171
ハードウェアベース, 171
プライマリ, 174
リングの割り当て, 175

N

NIC リング, 22, 22
参照 リング割り当て
受信と送信, 171

O

Oracle Solaris エラスティック仮想スイッチ, 99, 101
参照 EVS
Oracle Solaris カーネルゾーン, 117
Oracle VSI マネージャー, 84, 90
`oracle_v1` エンコーディング
定義, 91
`oracle_v1`エンコーディング, 84
`ORACLE_VSIMGR_V1`, 90

S

SCTP, 187
SLA
仮想スイッチ, 100
SR-IOV
VNIC, 53
仮想機能 *参照* VF
データリンクに対する有効化, 53
データリンクのチェック, 53
SSH 認証
設定, 125

T

TCP, 187

U

UDP, 187

V

VDP TLV, 90
VDP *参照* VSI Discovery and Configuration Protocol
VDP (VSI Discovery and Configuration Protocol)
Ethernet リンクの VDP 状態, 92
VDP TLV, 90
VDP 統計情報の表示, 93
VDP 統計情報、コマンドの表示, 93
VDP を使用した VNIC 情報の交換, 90
VF VNIC
移行, 56
作成, 54
表示, 57
VF VNIC の作成, 54
VF 情報
表示, 57
VF の表示, 57
VLAN, 101, 103
VNIC として, 28
VNIC の VLAN ID の変更, 45
ゾーンでの一時的な作成, 37
VM 間の通信を管理するための LLDP の使用, 88
VM での通信の制御, 85
VNI, 61
VNIC `anet` リソース, 110, 114
VNIC, 14, 15, 15
CPU プールリソースの割り当て, 184
MAC アドレスの変更, 46

SR-IOV の使用, 53
VLAN ID の変更, 45
VLAN ID を持つ, 28
VLAN として, 26
VNIC IP インタフェースの作成, 28
移行, 48
管理, 40
構成, 27, 31
削除, 50
作成, 25
システムで作成された, 26
情報の表示, 41
静的 IP アドレスの割り当て, 28
ゾーンでの一時的な作成, 36
ゾーンでの構成, 34
ゾーンでの使用, 33
ゾーンへの割り当て, 34
複数の MAC アドレスの表示, 42
プライベート仮想ネットワーク, 37
プロパティーの設定, 170
ベースとなるリンクの変更, 48
VNIC での EVB 関連のプロパティーの表示例, 98
VNIC の一時的な作成
ゾーンで, 36
VNIC の削除, 50
VPort
エラスティック仮想スイッチへの追加, 135
管理, 139, 145
削除, 150
表示, 148
プロパティーの設定, 146
プロパティーの表示, 146
リセット, 149
VPort のプロパティー, 104
IP アドレス, 104
MAC アドレス, 104
SLA
最大帯域幅, 104
サービスクラス, 104
優先順位, 104
VSI *参照* 仮想ステーションインスタンス
VSI 識別子, 90
VSI タイプ ID, 90
VSI バージョン, 90
VSI プロファイル, 90
VSI マネージャー, 90
`oracle_v1`, 90
VSI マネージャー ID
`ORACLE_VSIMGR_V1`, 90
VXLAN, 20, 60, 101, 103, 115
`anet` リソース, 64
VNIC, 64
VXLAN エンドポイント, 61
VXLAN セグメント, 64
VXLAN の構成, 67
VXLAN の作成例, 70
概要, 60
下位リンク, 72
構成の計画, 66
削除, 72
ゾーンとの使用, 64
ゾーンの `anet` への VXLAN の割り当ての例, 73
ゾーンへの割り当て, 72
ゾーン用の構成, 67
トポロジ, 62
表示, 72
命名規則, 61
要件, 66
利点, 60
VXLAN 上に作成された VNIC のゾーンへの割り当ての例, 71
VXLAN セグメント ID, 61, 66 *参照* VNI
VXLAN の削除, 72
VXLAN の表示, 72

Z

`zoneadm` コマンド, 32
`zonecfg` コマンド, 31, 110, 185, 194