



# Red Hat OpenStack Platform 16.2

## ネットワーク機能仮想化 (NFV) の製品ガイド

ネットワーク機能仮想化 (NFV) の概要



# Red Hat OpenStack Platform 16.2 ネットワーク機能仮想化 (NFV) の製品ガイド

---

## ネットワーク機能仮想化 (NFV) の概要

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## 法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Network\_Functions\_Virtualization\_Product\_Guide.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 概要

本ガイドでは、ネットワーク機能仮想化 (NFV)、その利点、サポートされる設定、アーキテクチャー、コンポーネント、インストール、統合に関する情報を紹介します。

## 目次

前書き .....	3
多様性を受け入れるオープンソースの強化 .....	4
RED HAT ドキュメントへのフィードバック (英語のみ) .....	5
第1章 RED HAT ネットワーク機能仮想化 (NFV) の理解 .....	6
1.1. NFV の利点 .....	6
1.2. NFV デプロイメントでサポートされている構成 .....	6
第2章 ソフトウェア .....	8
2.1. ETSI NFV アーキテクチャー .....	8
2.2. NFV ETSI のアーキテクチャーおよびコンポーネント .....	8
2.3. RED HAT NFV のコンポーネント .....	10
2.4. NFV インストールの概要 .....	10
第3章 NFV ハードウェア .....	12
第4章 NFV データプレーンの接続性 .....	13
4.1. 高速データパスのオプション .....	13
第5章 NFV のパフォーマンスの考慮事項 .....	15
5.1. CPU と NUMA ノード .....	15
5.1.1. NUMA ノードの例 .....	15
5.1.2. NUMA 対応のインスタンス .....	16
5.2. CPU ピニング .....	16
5.3. ヒュージページ .....	17
5.4. ポートセキュリティ .....	17
第6章 詳細情報 .....	18



# 前書き

## 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[弊社 の CTO、Chris Wright のメッセージ](#) を参照してください。



## RED HAT ドキュメントへのフィードバック (英語のみ)

弊社ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。

### ドキュメントへのダイレクトフィードバック (DDF) 機能の使用 (英語版のみ)

特定の文章、段落、またはコードブロックに対して直接コメントを送付するには、DDF の **Add Feedback** 機能を使用してください。なお、この機能は英語版のドキュメントでのみご利用いただけます。

1. **Multi-page HTML** 形式でドキュメントを表示します。
2. ドキュメントの右上隅に **Feedback** ボタンが表示されていることを確認してください。
3. コメントするテキスト部分をハイライト表示します。
4. **Add Feedback** をクリックします。
5. **Add Feedback** フィールドにコメントを入力します。
6. (オプション) ドキュメントチームが連絡を取り問題についてお伺いできるように、ご自分のメールアドレスを追加します。
7. **Submit** をクリックします。

## 第1章 RED HAT ネットワーク機能仮想化 (NFV) の理解

ネットワーク機能仮想化 (NFV: Network Functions Virtualization) とは、通信事業者 (CSP) が従来のプロプライエタリーハードウェアの範囲を超えて、運用コストを抑えつつ、効率性および俊敏性を向上させる上で役立つソフトウェアベースのソリューションのことです。

NFV 環境では、スイッチ、ルーター、ストレージなどの標準のハードウェアデバイス上で実行する標準の仮想化技術を使用して仮想化インフラストラクチャーを提供することで、IT およびネットワークの収束が可能になり、ネットワーク機能を仮想化 (VNF) します。管理およびオーケストレーションロジックにより、これらのサービスをデプロイおよび維持します。NFV には、手動操作の必要性を軽減するシステム管理、自動化、ライフサイクル管理も含まれます。

### 1.1. NFV の利点

ネットワーク機能仮想化 (NFV) を実装する主な利点は以下のとおりです。

- 新たなネットワークサービスを迅速にデプロイおよびスケールアップすることが可能なので、市場投入までの時間を短縮し、ニーズの変化に対応することができます。
- サービスの開発者は、実稼動環境で使用する場合と同じプラットフォームを使用して、リソースやプロトタイプを自己管理できるので、イノベーションがサポートされます。
- セキュリティやパフォーマンスを犠牲にせず、週/日単位ではなく、時間/分単位で顧客のニーズに対応します。
- カスタマイズされた高額な設備ではなく、商用オフザシェルフ (COTS) のハードウェアを使用するため、資本支出が削減されます。
- 運用の合理化と自動化で日常のタスクを最適化することにより、運用コストを削減し、従業員の生産性を向上させます。

### 1.2. NFV デプロイメントでサポートされている構成

Red Hat OpenStack Platform (RHOSP) では、OVS-DPDK および SR-IOV 設定の自動化が追加され、NFV デプロイメントがサポートされています。テクノロジープレビューと記した機能のサポート範囲についての詳細は、[「Technology Preview」](#) を参照してください。

#### ハイパーコンバージドインフラストラクチャー (HCI)

コンピュートサブシステムと Red Hat Ceph Storage ノードを共存させることができます。NFV ユースケースにおいて、このハイパーコンバージドモデルを使用すると、初期コストの軽減、初期デプロイメントのフットプリントの縮小、稼働率の最大化、管理の効率化などが実現されます。HCI の詳細は、[「Hyperconverged Infrastructure Guide」](#) を参照してください。

#### コンポーザブルロール

コンポーザブルロールを使用して、カスタムのデプロイメントを作成できます。コンポーザブルロールを使用して、各ロールにサービスを追加/削除することができます。コンポーザブルロールの詳細は、[「Composable Roles and Services」](#) を参照してください。

#### Open vSwitch (OVS) と LACP の組み合わせ

OVS 2.9 では、LACP と OVS の組み合わせが完全にサポートされています。この構成は、Openstack のコントロールプレーンのトラフィックには推奨されません。OVS または Openstack Networking で機能の中断が発生すると、管理機能が損なわれるためです。詳細は、[「Open vSwitch Bonding Options」](#) を参照してください。

#### OVS ハードウェアオフロード

Red Hat OpenStack Platform では、一部制約はあるものの、OVS ハードウェアオフロードのデプロイメントをサポートしています。ハードウェアオフロードに対応した OVS のデプロイに関する詳細は、[「OpenvSwitch Hardware offload」](#) を参照してください。

### Open Virtual Network (OVN)

RHOSP 16.1.4 では、以下の NFV OVN の構成を使用することができます。

- [OVN with OVS-DPDK colocated with SR-IOV](#)
- [OVN with OVS TC Flower offload](#)

## 第2章 ソフトウェア

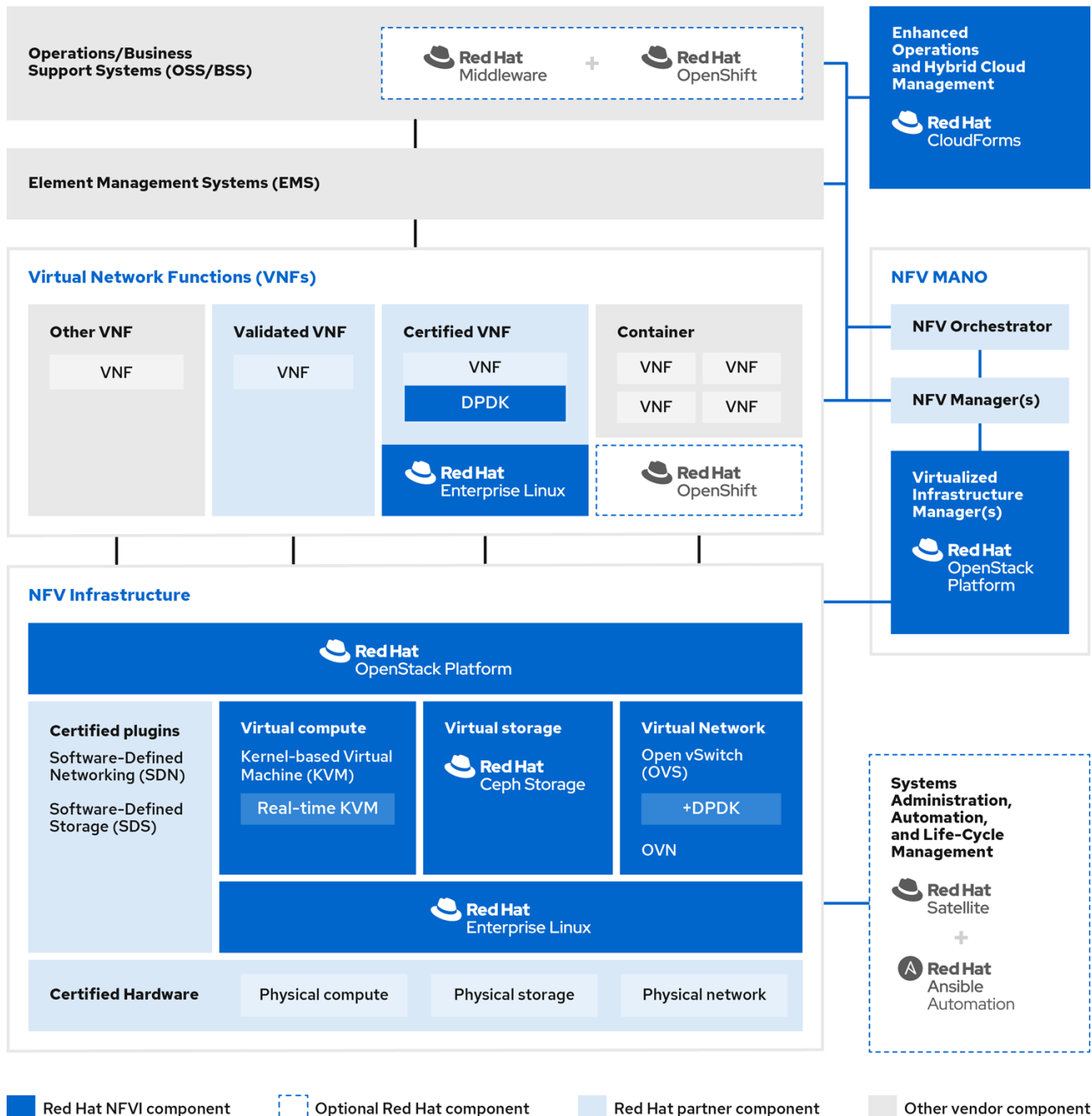
### 2.1. ETSI NFV アーキテクチャー

欧州電気通信標準化機構 (ETSI) は、ヨーロッパの情報通信技術 (ICT: Information and Communication Technology) の標準を開発する独立した標準化組織です。

ネットワーク機能仮想化 (NFV) は、プロプライエタリーのハードウェアデバイスの使用に伴う問題への対処に重点を置いています。NFV を使用すると、ユースケースの要件や経済的な利点によっては、ネットワーク固有の設備をインストールする必要性が軽減されます。ETSI Industry Specification Group for Network Functions Virtualization (ETSI ISG NFV) は、VF を確実にサポートするために必要な要件、リファレンスアーキテクチャー、インフラストラクチャー仕様を設定します。

Red Hat では、通信事業者 (CSP) の IT とネットワークのコンバージェンス実現を支援するオープンソースベースのクラウド最適化ソリューションを提供しています。Red Hat は、Single Root I/O Virtualization (SR-IOV) や Open vSwitch with Data Plane Development Kit (OVS-DPDK) などの NFV 機能を Red Hat OpenStack に追加します。

### 2.2. NFV ETSI のアーキテクチャーおよびコンポーネント



140\_OpenStack\_0221

通常、ネットワーク機能仮想化 (NFV) プラットフォームには、以下のコンポーネントが含まれます。

- **仮想ネットワーク機能 (VNF):** ルーター、ファイアウォール、ロードバランサー、ブロードバンドのゲートウェイ、モバイルパケットのプロセッサ、サービス提供ノード、シグナリング、位置情報サービスなどのネットワーク機能のソフトウェア実装。
- **ネットワーク機能仮想化インフラストラクチャー (NFVi):** インフラストラクチャーを構成する物理リソース (コンピュータ、ストレージ、ネットワーク) および仮想化層。このネットワークには、仮想マシン間、ホスト全体でパケットを転送するためのデータパスが含まれます。これにより、基盤のハードウェアの情報を考慮せずに VNF をインストールできます。NFVi は、NFV スタックの基盤を形成します。NFVi は、マルチテナントをサポートし、Virtual Infrastructure Manager (VIM) で管理されます。Enhanced Platform Awareness (EPA) は低レベルの CPU および NIC アクセラレーション機能を VNF に公開し、仮想マシンのパケット転送のパフォーマンス (スループット、レイテンシー、ジッター) を向上させます。

- **NFV Management and Orchestration (MANO)** VNF のライフサイクル全体で必要とされる全サービス管理タスクにフォーカスする管理およびオーケストレーション層。MANO の主要な目的は、オペレーターが顧客に提供するネットワーク機能のサービス定義、自動化、エラーの相関関係、監視、ライフサイクル管理を物理インフラストラクチャーから切り離せるようにすることです。このような切り離しを行うには、Virtual Network Function Manager (VNFM) が提供する管理層が追加で必要になります。VNFM は、直接対話するか、VFN ベンダーが提供する Element Management System (EMS) を使用して、仮想マシンのライフサイクルや VNF を管理します。MANO が定義するコンポーネントでもう 1 つ重要なのは、NFVO として知られるオーケストレーターです。NFVO は、最上部のオペレーション/ビジネスサポートシステム (OSS/BSS) や、最下部の VNFM など、さまざまなデータベースやシステムにインターフェースを提供します。NFVO は、顧客向けの新規サービスを構築する場合、VNFM に対して VNF のインスタンス化をトリガーするよう頼みます (これにより、複数の仮想マシンが作成される場合があります)。
- **オペレーション/ビジネスサポートシステム (OSS/BSS: Operations/Business Support System):** オペレーションサポートや請求など必要不可欠なビジネス機能アプリケーションを提供します。OSS/BSS は、NFV に適応する必要がある、レガシーシステムと新規の MANO コンポーネントを統合しています。BSS システムは、サービスサブスクリプションをベースにポリシーを設定して、レポートと請求を管理します。
- **システム管理、自動化、ライフサイクル管理:** システム管理、インフラストラクチャーコンポーネントの自動化、および NFVi プラットフォームのライフサイクルを管理します。

## 2.3. RED HAT NFV のコンポーネント

Red Hat の NFV 向けのソリューションには、ETSI モデルに含まれる NFV フレームワークの異なるコンポーネントとしての役割を果たすことのできる各種製品が含まれます。NFV ソリューションには、Red Hat ポートフォリオの以下の製品が統合されます。

- Red Hat OpenStack Platform: IT および NFV ワークロードをサポートします。Enhanced Platform Awareness (EPA) 機能は、SR-IOV や OVS-DPDK をサポートする CPU ピニング、ヒュージページ、Non-Uniform Memory Access (NUMA) アフィニティー、ネットワークアダプター (NIC) などを使用して、決定的なパフォーマンスの向上を図ることができます。
- Red Hat Enterprise Linux および Red Hat Enterprise Linux Atomic Host: VNF として仮想マシンやコンテナを作成します。
- Red Hat Ceph Storage: サービスプロバイダーのワークロードのすべてのニーズに対応する弾力性があり、統一された高性能なストレージ層を提供します。
- Red Hat JBoss Middleware および Red Hat 提供の OpenShift Enterprise: オプションで OSS/BSS コンポーネントの最新化に使用することができます。
- Red Hat CloudForms: VNF マネージャーを提供し、統合された画面に、VIM や NFVI などの複数のソースのデータを表示することができます。
- Red Hat Satellite および Red Hat 提供の Ansible: オプションでシステムの管理、自動化、ライフサイクル管理を強化します。

## 2.4. NFV インストールの概要

Red Hat OpenStack Platform director は完全な OpenStack 環境をインストールおよび管理します。director は、アップストリームの OpenStack TripleO プロジェクトをベースとしています。TripleO とは、「OpenStack-On-OpenStack」の頭文字の「O」が 3 つあることを意味しています。このプロジェ

クトは、OpenStack コンポーネントを活用して、完全に機能する OpenStack 環境をインストールします。director は、リーンかつ堅牢な Red Hat OpenStack Platform の完全な環境を簡単にインストールできる方法を提供します。

アンダークラウドおよびオーバークラウドのインストールに関する詳細は、『[Red Hat OpenStack Platform Director Installation and Usage](#)』を参照してください。

NFV 機能をインストールするには、以下の手順を実施します。

- **network-environment.yaml** ファイルへの SR-IOV および PCI パススルーのパラメーターの追加、CPU チューニング用 **post-install.yaml** ファイルの更新、**compute.yaml** ファイルの変更、そして **overcloud\_deploy.sh** スクリプトの実行により、オーバークラウドをデプロイします。
- NIC から直接データをポーリングして、高速パケット処理用の DPDK ライブラリーおよびドライバーをインストールします。**network-environment.yaml** ファイルへの DPDK パラメーターの追加、CPU チューニング用 **post-install.yaml** ファイルの更新、**compute.yaml** ファイルの更新による DPDK ポートでのブリッジの設定、**controller.yaml** ファイルの更新によるブリッジと VLAN が設定されたインターフェースの設定、そして **overcloud\_deploy.sh** の実行により、オーバークラウドをデプロイします。

必要な NFV のプランニングガイドラインと設定についての詳細は、『[Network Function Virtualization Planning and Configuration Guide](#)』を参照してください。

## 第3章 NFV ハードウェア

OpenStack ノードのハードウェア選択に関する考慮事項は、[『Director Installation and Usage』](#) を参照してください。

ネットワーク機能仮想化 (NFV) 向けのテスト済み NIC の一覧は、[「Network Adapter Fast Datapath Feature Support Matrix」](#) を参照してください (カスタマーポータルログインが必要です)。



## 第4章 NFV データプレーンの接続性

NFV が導入され、従来のデバイスを VNF として実装を始めているネットワークベンダーが増えています。これらのネットワークベンダーの大半が仮想マシンに着目していますが、選択した設計に合わせたコンテナベースのアプローチに注目しているベンダーもあります。OpenStack ベースのソリューションは、主に以下の2つの理由からリッチかつ柔軟でなければなりません。

- **アプリケーションの即応性:** ネットワークベンダーは現在、デバイスを VNF に変換している段階にあります。VNF 市場では、VNF ごとに成熟度レベルは異なり、即応性に関する共通の障害として、API での RESTful インターフェースの有効化、データモデルのステートレスへの進化、自動化管理オペレーションの提供などが挙げられます。OpenStack は、すべてに共通のプラットフォームを提供する必要があります。
- **幅広いユースケース:** NFV には、異なるユースケースに対応する多様なアプリケーションが含まれます。たとえば、Virtual Customer Premise Equipment (vCPE) は、ルーティング、ファイアウォール、仮想プライベートネットワーク (VPN)、ネットワークアドレス変換 (NAT) など多くのネットワーク機能を顧客のサイトで提供することを目的としています。Virtual Evolved Packet Core (vEPC) は、Long-Term Evolution (LTE) ネットワークのコアコンポーネントにコスト効果の高いプラットフォームを提供するクラウドアーキテクチャーで、ゲートウェイやモバイルエンドポイントをダイナミックにプロビジョニングして、スマートフォンやその他のデバイスからの増加するデータトラフィック量に対応できるようにします。これらのユースケースは、異なるネットワークアプリケーションとプロトコルを使用して実装され、インフラストラクチャーとは異なる接続性、分離、パフォーマンスの特性を必要とします。また、コントロールプレーンのインターフェースとプロトコル、実際の転送プレーンを分離することが一般的です。OpenStack には、さまざまなデータパスの接続性オプションを提供できるように十分な柔軟性が必要です。

基本的に、仮想マシンにデータプレーンの接続性を提供する一般的なアプローチは2種類あります。

- **ハードウェアへのダイレクトアクセス:** PCI パススルー (OpenStack では SR-IOV PF と表記) や、VF (Virtual Function) と PF (Physical Function) の両パススルー向けの Single Root I/O Virtualization (SR-IOV) などの技術を使用し、Linux カーネルを迂回して物理ネットワークインターフェースカード (NIC) にセキュアな Direct Memory Access (DMA) を提供します。
- **仮想スイッチ (vswitch) の使用:** ハイパーバイザーのソフトウェアサービスとして実装されています。仮想マシンは、仮想インターフェース (vNIC) を使用して vSwitch に接続され、vSwitch は仮想マシン間や仮想マシンと物理ネットワーク間のトラフィックを転送することができます。

### 4.1. 高速データパスのオプション

高速データパスには、以下のようなオプションがあります。

- **Single Root I/O Virtualization (SR-IOV):** 単一の PCI ハードウェアデバイスを複数の仮想 PCI デバイスのように見せる標準のことです。これは、Physical Function (PF) を導入して機能します。PF とは、物理ハードウェアポートと Virtual Function (VF: 仮想マシンに割り当てられた軽量機能) の機能を果たすフル装備の PCIe 機能です。仮想マシンは、VF をハードウェアと直接通信する通常の NIC とみなします。NIC は複数の VF をサポートします。
- **Open vSwitch (OVS):** 仮想化サーバー環境内で仮想スイッチとして使用されるように設計されたオープンソースのソフトウェアスイッチです。OVS は、通常の L2-L3 スイッチのケーパビリティだけでなく、ユーザー定義のオーバーレイネットワーク (例: VXLAN) を作成する OpenFlow などの SDN プロトコルもサポートします。OVS は、物理 NIC を使用する仮想マシン間およびホスト全体のパケットの切り替えに、Linux のカーネルネットワークを使用します。OVS は、iptables/ebtables で Linux ブリッジのオーバーヘッドを回避する接続トラッキング

グ (Conntrack) と内蔵のファイアウォール機能をサポートするようになりました。Red Hat OpenStack Platform 環境の Open vSwitch は、カスタマイズなしに OVS と OpenStack Networking (neutron) を統合できます。

- **Data Plane Development Kit (DPDK):** 高速なパケット処理に向けてライブラリーセットや Poll Mode Driver (PMD) で構成されます。DPDK はユーザー空間で大半を実行するように設計されており、アプリケーションが NIC から (または NIC へ) 直接独自のパケット処理を実行できるようになります。DPDK は、レイテンシーを減らし、処理するパケット数を増やすことができます。DPDK Poll Mode Drivers (PMDs) はビジーループで実行され、ホストの NIC ポートやゲストの vNIC ポートにパケットが到着していないかを絶えずスキャンします。
- **DPDK accelerated Open vSwitch (OVS-DPDK):** Linux カーネルバイパスと物理 NIC への Direct Memory Access (DMA) を用いた高性能のユーザー空間ソリューションに向け DPDK がバンドルされた Open vSwitch。これは、標準の OVS カーネルデータパスを DPDK ベースのデータパスに置き換えて、内部で DPDK をパケット転送に使用するユーザー空間の vSwitch をホストに構築するという発想になります。このアーキテクチャーの利点は、大半がユーザーに透過的である点です。OpenFlow、OVSDB、コマンドラインなどの公開されるインターフェースは、ほぼ同じ状態のままになります。

## 第5章 NFV のパフォーマンスの考慮事項

ネットワーク機能仮想化 (NFV) ソリューションが有用であるためには、VF が物理実装のパフォーマンス以上である必要があります。Red Hat の仮想化技術は、OpenStack およびクラウドのデプロイメントで一般的に使用されている高パフォーマンスの Kernel-based Virtual Machine (KVM) ハイパーバイザーをベースにしています。

### 5.1. CPU と NUMA ノード

以前は、x86 システムの全メモリーは、システム内のどの CPU からでも同等にアクセスできていました。これにより、システム内で操作を行う CPU、Uniform Memory Access (UMA) を参照する CPU はどれでも、メモリーのアクセス時間が同じでした。

Non-Uniform Memory Access (NUMA) では、システムメモリーは、特定の CPU またはソケットに割り当てられるノードと呼ばれるゾーンに分割されます。CPU のローカルにあるメモリーには、そのシステムのリモートの CPU に接続されているメモリーにアクセスするよりも高速です。通常、NUMA システム上のソケットにはそれぞれローカルのメモリーノードがあり、別の CPU のローカルにあるノードのメモリーや、全 CPU で共有されるバス上のメモリーよりも、コンテンツに早くアクセスできます。

同様に、物理 NIC はコンピュータノードのハードウェア上の PCI スロットに配置されます。これらのスロットは、特定の NUMA ノードに関連付けられる特定の CPU ソケットに接続されます。パフォーマンスを最適化するには、CPU の設定 (SR-IOV または OVS-DPDK) と同じ NUMA ノードにデータパス NIC を接続します。

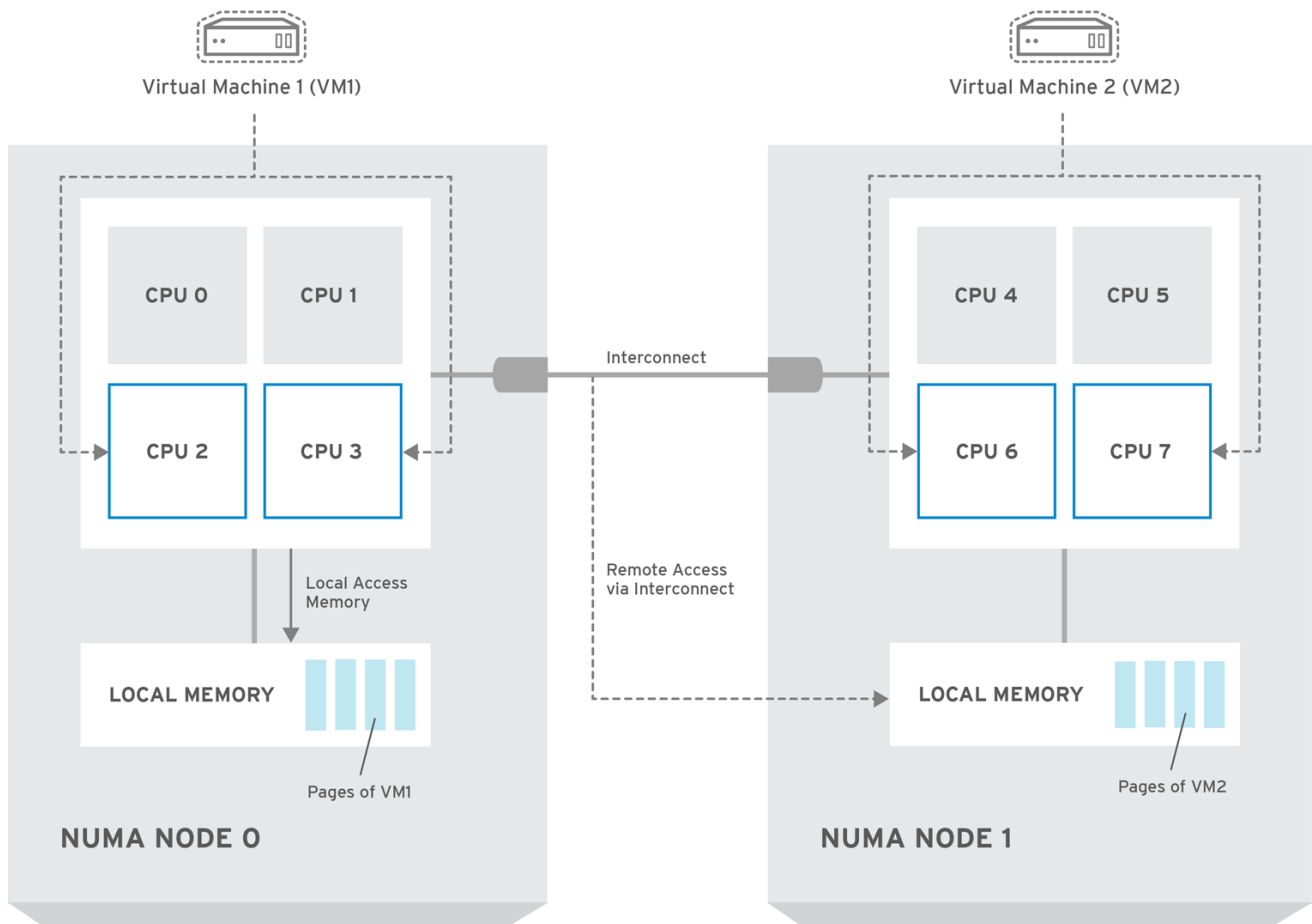
NUMA を使用しない場合のパフォーマンスへの影響は大きく、一般的に、パフォーマンスの10%以上が影響を受けます。各 CPU ソケットには、仮想化を目的とする個別の CPU として扱われる複数の CPU コアを配置することができます。

#### ヒント

NUMA の詳細は、[「What is NUMA and how does it work on Linux?」](#) を参照してください。

#### 5.1.1. NUMA ノードの例

以下の図では、2つのノードからなる NUMA システムの例と、CPU コアとメモリーページが利用できる方法の例を示しています。



OPENSTACK\_39825\_0516



## 注記

Interconnect 経由で利用可能なリモートのメモリーには、NUMA ノード 0 からの VM1 に NUMA ノード 1 の CPU コアがある場合 **のみ**、アクセスすることができます。この場合、NUMA ノード 1 のメモリーは、VM1 の 3 番目の CPU コアのローカルとして機能します (例: 上記の図では VM1 は CPU4 が割り当てられています) が、それと同時に、同じ仮想マシンの別の CPU コアに対してはリモートメモリーとして機能します。

### 5.1.2. NUMA 対応のインスタンス

NUMA アーキテクチャーを持つシステム上で NUMA トポロジー認識を使用するように OpenStack 環境を設定することができます。仮想マシン (VM) でゲストオペレーティングシステムを実行する場合は、NUMA トポロジーが 2 つあります。

- ホストの物理ハードウェアの NUMA トポロジー
- ゲストオペレーティングシステムに公開される仮想ハードウェアの NUMA トポロジー

仮想ハードウェアを物理ハードウェア NUMA トポロジーに合わせて調整することで、ゲストオペレーティングシステムのパフォーマンスを最適化することができます。

## 5.2. CPU ピニング

CPU ピニングは、指定のホスト内にある特定の物理 CPU 上で特定の仮想マシンの仮想 CPU を実行する機能のことです。vCPU ピニングでは、ベアメタルシステムへのピンングタスクと同様の利点が得られます。仮想マシンは、ホストのオペレーティングシステムのユーザー空間タスクとして実行されるの

で、ピンングすることでキャッシュの効率性が向上します。

CPU ピンニングの設定方法の詳細は、『[Configuring the Compute Service for Instance Creation](#)』の「[Configuring CPU pinning on the Compute node](#)」を参照してください。

### 5.3. ヒュージページ

物理メモリーは、ページと呼ばれる連続した一連のリージョンに分割されます。効率化を図るため、システムは、各メモリーバイトにアクセスするのではなく、ページ全体にアクセスしてメモリーを取得します。このような変換を実行するには、システムは、最近使用されたページまたは頻繁に使用されるページの物理から仮想アドレスへのマッピングが含まれるトランスレーションルックアサイドバッファ (TLB: Translation Lookaside Buffers) をチェックします。検索したマッピングが TLB にはない場合には、プロセッサは全ページテーブルで同じ処理を反復して、アドレスマッピングを判断する必要があります。TLB を最適化し、これらの TLB ミス時に発生するパフォーマンスペナルティーを最小限に抑えます。

X86 システムの通常のページサイズは 4 KB ですが、他の大きなページサイズを利用することもできます。ページサイズが大きいと全体的なページ数が少なくなるので、TLB に仮想から物理アドレスへの変換を保管可能なシステムメモリー量が増えることとなります。その結果、TLB ミスの可能性が低くなり、パフォーマンスが向上します。ページサイズが大きいと、プロセスはページに割り当てる必要があるため、メモリーを無駄にする可能性が高くなりますが、すべてのメモリーが必要となることはあまりありません。そのため、ページサイズを選択する際には、より大きいページを使用してアクセス時間を速くするか、より小さいページを使用して最大限にメモリーが使用されるようにするかで、トレードオフが生じます。

### 5.4. ポートセキュリティー

ポートセキュリティーは、不正なアクセスを防ぐ手段です。発信元ネットワークポートのソース IP およびソース MAC アドレスと一致しない送信トラフィックをブロックします。セキュリティーグループルールを使用して、この挙動を監視または変更することはできません。

デフォルトでは、OpenStack で新たに作成される Neutron ネットワークの **port\_security\_enabled** パラメーターは、**enabled** に設定されます。ネットワーク上で新たに作成されるポートは、そのネットワークから **port\_security\_enabled** パラメーターの値をコピーします。

ファイアウォールまたはルーターの構築など一部の NFV のユースケースでは、ポートセキュリティーを無効にしなければならない場合があります。

特定のポートでポートセキュリティーを無効にするには、以下のコマンドを実行します。

```
openstack port set --disable-port-security <port-id>
```

ネットワークで作成されるすべての新規ポートで、ポートセキュリティーの有効化を阻止するには、以下のコマンドを実行します。

```
openstack network set --disable-port-security <network-id>
```

## 第6章 詳細情報

以下の表には、その他の Red Hat の参照資料一覧を記載しています。

Red Hat OpenStack Platform のドキュメントスイートは、[Red Hat OpenStack Platform の製品ドキュメント](#) を参照してください。

表6.1 参照資料一覧

コンポーネント	参考情報
Red Hat Enterprise Linux	Red Hat OpenStack Platform は Red Hat Enterprise Linux 8.0 でサポートされています。Red Hat Enterprise Linux のインストールに関する情報は、 <a href="#">Red Hat Enterprise Linux の製品ドキュメント</a> で対応するインストールガイドを参照してください。
Red Hat OpenStack Platform	<p>OpenStack のコンポーネントとそれらの依存関係をインストールするには、Red Hat OpenStack Platform director を使用します。director は、基本的な OpenStack 環境を <b>アンダークラウド</b> として使用して、最終的な <b>オーバークラウド</b> 内の OpenStack ノードをインストール、設定、管理します。アンダークラウドのインストールには、デプロイしたオーバークラウドに必要な環境に加えて、アンダークラウドのインストールには、追加のホストマシンが必要となる点に注意してください。詳しい手順は、<a href="#">『Red Hat OpenStack Platform Director Installation and Usage』</a> を参照してください。</p> <p>ネットワークの分離、ストレージ設定、SSL 通信など、Red Hat OpenStack Platform director を使用して Red Hat OpenStack Platform エンタープライズ環境に高度な機能を設定する方法、および一般的な設定方法については、<a href="#">『Advanced Overcloud Customization』</a> を参照してください。</p>
NFV のドキュメント	Single Root I/O virtualization (SR-IOV) および Open vSwitch with Data Plane Development Kit (OVS-DPDK) を使用した Red Hat OpenStack Platform デプロイメントのプランニングおよび設定に関する詳細は、 <a href="#">『ネットワーク機能仮想化 (NFV) のプランニングおよび設定ガイド』</a> を参照してください。