

Cisco UCS B シリーズ ブレード サーバの Cisco usNIC 展開ガイド

初版 : 2016 年 01 月 20 日

Cisco usNIC の概要

Cisco user-space NIC (Cisco usNIC) 機能は、ネットワーキング パケットを送受信するときにカーネルをバイパスすることで、データセンターの Cisco UCS サーバで実行されるソフトウェア アプリケーションのパフォーマンスを改善します。アプリケーションは、第二世代以降の Cisco UCS VIC アダプタと直接やり取りすることによって、ハイ パフォーマンス コンピューティング クラスターのネットワーキングパフォーマンスが向上します。Cisco usNIC のメリットを引き出すためには、アプリケーションはソケットまたはその他の通信 API ではなく、Message Passing Interface (MPI) または Libfabric インターフェイスを使用する必要があります。

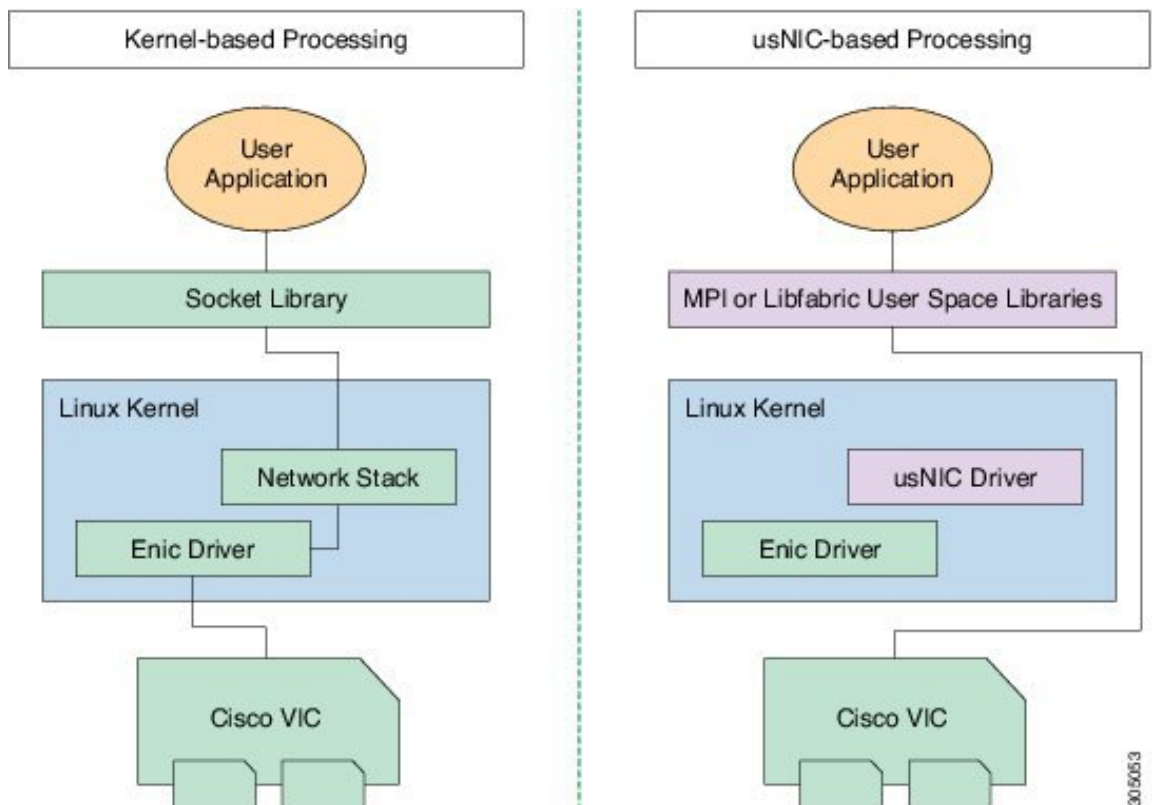
Cisco usNIC は、お使いのアプリケーションに対して次の利点を提供します。

- 低遅延で、高スループットの通信転送を提供します。
- 標準のアプリケーション非依存イーサネット プロトコルを実行します。
- 低ジッター、またはほぼ一定の遅延の通信。
- 次に示すシスコ データセンター プラットフォームで、低遅延の転送、ユニファイド ファブリック、統合管理のサポートを活用します。
 - Cisco UCS サーバ
 - 第二世代以降の Cisco UCS VIC アダプタ
 - 10 または 40GbE ネットワーク

標準イーサネット アプリケーションは、Linux カーネルのネットワーキング スタックを呼び出すユーザ領域のソケット ライブラリを使用します。次に、ネットワーキング スタックは Cisco eNIC

ドライバを使用して、Cisco VIC ハードウェアと通信します。次の図は、通常のソフトウェアアプリケーションと usNIC を使用する MPI アプリケーションの対比を示します。

図 1：カーネルベースのネットワーク通信と Cisco usNIC ベースの通信



Cisco usNIC の前提条件

Cisco usNIC から恩恵を受けるためには、設定に次の前提条件があります。

- UCS Driver ISO。詳細については、『[Downloading Cisco UCS VIC drivers](#)』を参照してください。
- サポートされている Linux オペレーティング システム ディストリビューション リリース。詳細については、該当する『[Hardware and Software Interoperability guide](#)』を参照してください。
- Cisco Open MPI ディストリビューションなどのサポート対象の MPI 実装（Cisco UCS Driver ISO を含む）または Intel® MPI ライブラリのバージョン 4 または 5。Intel® MPI ライブラリが使用されている場合は、ネットワークでフロー制御を有効に設定する必要があります。

Cisco usNIC の設定



(注) Cisco usNIC パッケージは、オペレーティング システムのアップグレードまたはダウングレードをサポートしません。オペレーティング システムを更新するには、まず、usNIC パッケージをアンインストールして、オペレーティング システムを更新してから、usNIC ドライバを再インストールします。

または、オペレーティング システムを更新して、usNIC ドライバをアンインストールしてから、usNIC ドライバを再インストールすることができます。

はじめる前に

次のソフトウェア コンポーネントとハードウェア コンポーネントが Cisco UCS サーバにインストールされていることを確認してください。

- サポートされている Linux オペレーティング システム ディストリビューション リリース。詳細については、該当する『[Hardware and Software Interoperability guide](#)』を参照してください。
- GCC、G++、および Gfortran
- DAT ユーザ ライブラリ (Intel® MPI を使用している場合)
- libnl ユーザ ライブラリ (バージョン 1 とバージョン 3 のどちらか)
- 第二世代以降の Cisco UCS VIC アダプタ



重要

サポートされている Linux オペレーティング システム ディストリビューションについては、UCS Drivers ISO バンドルに含まれている usNIC フォルダの内容を参照してください。『[Cisco UCS Virtual Interface Card Drivers for Linux Installation Guide](#)』を参照してください。

手順の概要

1. カーネルでカーネル オプション **CONFIG_INTEL_IOMMU** が選択されていることを確認します。手動で grub.conf ファイル (たとえば、/boot/grub/grub.conf) に「intel_iommu=on」を追加することによって、Linux カーネル内で Intel IOMMU ドライバを有効にします。
2. Cisco UCS サーバをリブートします。
3. 実行中のカーネルが intel_iommu=on オプションでブートすることを確認します。
4. Cisco usNIC の Linux ドライバをインストールします。

手順の詳細

ステップ 1 カーネルでカーネル オプション **CONFIG_INTEL_IOMMU** が選択されていることを確認します。手動で `grub.conf` ファイル (たとえば、`/boot/grub/grub.conf`) に `intel_iommu=on` を追加することによって、Linux カーネル内で Intel IOMMU ドライバを有効にします。

たとえば、`kernel (hd0,0)/vmlinuz LANG=en_US.UTF-8 KEYTABLE=us` のように、`grub.conf` ファイルに `kernel` 行が含まれている場合、`intel_iommu=on` を次に示すように最後に追加します。

```
kernel (hd0,0)/vmlinuz LANG=en_US.UTF-8 KEYTABLE=us intel_iommu=on
```

ステップ 2 Cisco UCS サーバをリブートします。
Intel IOMMU を有効にした後で、変更を反映するためにサーバを再起動します。

ステップ 3 実行中のカーネルが `intel_iommu=on` オプションでブートすることを確認します。

```
$ cat /proc/cmdline | grep iommu
```

ステップ 4 Cisco usNIC の Linux ドライバをインストールします。
ドライバのインストールの詳細については、[Cisco usNIC 用の Linux ソフトウェアパッケージのインストール](#)を参照してください。

次の作業

Cisco usNIC の設定と Linux ドライバのインストールが完了したら、Cisco usNIC が正しく動作していることを確認してください。インストールの確認方法の詳細については、[Cisco UCS B シリーズ ブレード サーバ向け Cisco usNIC 取り付けの確認](#)、(16 ページ) を参照してください。

Cisco Manager GUI を使用した Cisco usNIC 接続ポリシーの作成

Cisco usNIC 接続ポリシーを作成するには、後述する手順を使用することも、この[ビデオ](#)の [Play] をクリックしてやり方を確認することもできます。

手順の概要

1. [Navigation] ペインの [LAN] タブをクリックします。
2. [LAN] タブの [LAN] > [Policies] を展開します。
3. [root] ノードを展開します。
4. [usNIC Connection Policies] を右クリックして、[Create usNIC Connection Policy] を選択します。
5. [Create usNIC Connection Policy] ダイアログボックスで、次のフィールドに値を入力します。

手順の詳細

- ステップ 1 [Navigation] ペインの [LAN] タブをクリックします。
- ステップ 2 [LAN] タブの [LAN] > [Policies] を展開します。
- ステップ 3 [root] ノードを展開します。
- ステップ 4 [usNIC Connection Policies] を右クリックして、[Create usNIC Connection Policy] を選択します。
- ステップ 5 [Create usNIC Connection Policy] ダイアログボックスで、次のフィールドに値を入力します。

名前	説明
[Name] フィールド	ポリシーの名前。 この名前には、1～16文字の英数字を使用できます。- (ハイフン)、_ (アンダースコア)、: (コロン)、および . (ピリオド) は使用できますが、それ以外の特殊文字とスペースは使用できません。また、オブジェクトが保存された後にこの名前を変更することはできません。
[Description] フィールド	ポリシーの説明。ポリシーを使用する場所とタイミングに関する情報を含めることをお勧めします。
[Number of usNICs] フィールド	作成する usNIC の数。 サーバ上で動作している MPI プロセスごとに専用の usNIC が必要です。116 の MPI プロセスの同時実行を維持するために、1つのアダプタ上で最大 116 の usNIC を作成できます。usNIC 対応 vNIC ごとにサーバ上の物理コアと同数以上の usNIC を作成することをお勧めします。たとえば、サーバに 8つの物理コアがある場合は、8つの usNIC を作成します。
[Adapter Policy] ドロップダウンリスト	usNIC に指定するアダプタポリシー。デフォルトで作成される usNIC アダプタ ポリシーを選択することをお勧めします。

usNIC イーサネット アダプタ ポリシーの設定

ステップ 1 [Navigation] ペインの [Servers] タブをクリックします。

ステップ 2 [Servers] タブで、[Servers] > [Policies] > [root] > [Adapter Policies] の順に展開します。

ステップ 3 [Eth Adapter Policy usNIC] をクリックします。

ステップ 4 [Work] ペインで、[General] タブをクリックします。

必要に応じて、[Resources] セクションと [Options] セクションで詳細を変更できます。シスコでは、リソースに次のデフォルト値を使用することをお勧めしています。

名前	値
Transmit Queues	6
Ring Size	256
Received Queues	6
Ring Size	512
Completion Queues	6
Interrupts	6

Cisco UCS Manager GUI を使用した usNIC の変更

-
- ステップ 1 [Navigation] ペインの [Servers] タブをクリックします。
 - ステップ 2 [Servers] タブで、[Servers] > [Service Profiles] > ルートを展開します。
 - ステップ 3 usNIC を設定するサービス プロファイル ノードを展開して、[vNICs] をクリックします。
 - ステップ 4 [Work] ペインで、[Network] タブをクリックします。
 - ステップ 5 [vNICs] 領域で、vNIC を選択して、[Modify] をクリックします。
 - ステップ 6 [Modify vNIC] ダイアログボックスの [Adapter Performance Profile] 領域で、[Adapter Policy] ドロップダウン リストから [Linux] を選択します。
 - ステップ 7 [Connection Policies] 領域で、[usNIC] オプション ボタンをクリックします。
 - ステップ 8 [usNIC Connection Policy] ドロップダウン リストから、作成した usNIC 接続ポリシーを選択します。
 - ステップ 9 [OK] をクリックします。
 - ステップ 10 [Save Changes] をクリックします。
 - ステップ 11 [Navigation] ペインで、変更したばかりのサービス プロファイルをクリックします。
 - ステップ 12 [Work] ペインの [Policies] タブをクリックします。
 - ステップ 13 [BIOS Policy] バーを展開して、[BIOS Policy] ドロップダウン リストで [usNIC] を選択します。
 - ステップ 14 [Save Changes] をクリックします。
-

Cisco UCS Manager CLI を使用した usNIC の作成

はじめる前に

このタスクを実行するには、admin 権限を持つユーザとしてログインする必要があります。

手順の概要

1. UCS-A # **scope service-profile serverchassis-id/blade-id or rack_server-id**
2. UCS-A /org/service-profile # **show vnic**
3. UCS-A /org/service-profile # **scope vnicvnic name**
4. UCS-A /org/service-profile/vnic # **set adapter-policyLinux**
5. UCS-A /org/service-profile/vnic # **enter usnic-conn-policy-ref usnic connection policy reference name**
6. UCS-A /org/service-profile/vnic/usnic-conn-policy-ref* # **commit-buffer**
7. UCS-A /org/service-profile/vnic/usnic-conn-policy-ref # **top**
8. UCS-A # **scope org**
9. UCS-A /org # **create usnic-conn-policyusnic connection policy name**
10. UCS-A /org/usnic-conn-policy* # **set usnic-count number of usnics**
11. UCS-A /org/usnic-conn-policy* # **set adaptor-profile usNIC**
12. UCS-A /org/usnic-conn-policy* # **commit-buffer**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	UCS-A # scope service-profile serverchassis-id/blade-id or rack_server-id	指定されたシャーシ、ブレード、または UCS 管理ラック サーバ ID のサービスプロファイルを入力します。
ステップ 2	UCS-A /org/service-profile # show vnic	サーバ上で使用可能な vnic が表示されます。usNIC vNIC は Cisco UCS Manager リリース 2.2 にアップグレードするとデフォルトで使用できます。
ステップ 3	UCS-A /org/service-profile # scope vnicvnic name	指定された vNIC の vNIC モードを開始します。
ステップ 4	UCS-A /org/service-profile/vnic # set adapter-policyLinux	Linux と、usNIC 用のアダプタ ポリシーを指定します。
ステップ 5	UCS-A /org/service-profile/vnic # enter usnic-conn-policy-ref usnic connection policy reference name	vNIC 用の usNIC 接続ポリシー参照を指定された名前で作成します。接続ポリシー名の最大長は 16 文字です。
ステップ 6	UCS-A /org/service-profile/vnic/usnic-conn-policy-ref* # commit-buffer	トランザクションをシステムの設定にコミットします。
ステップ 7	UCS-A /org/service-profile/vnic/usnic-conn-policy-ref # top	最上位モードを開始します。
ステップ 8	UCS-A # scope org	ルート組織モードを開始します。
ステップ 9	UCS-A /org # create usnic-conn-policyusnic connection policy name	usNIC 接続ポリシーを指定された名前で作成します。

	コマンドまたはアクション	目的
ステップ 10	UCS-A /org/usnic-conn-policy* # set usnic-count <i>number of usnics</i>	作成する Cisco usNIC の数を指定します。この値に対して 58 を入力することをお勧めします。
ステップ 11	UCS-A /org/usnic-conn-policy* # set adaptor-profile usNIC	usNIC 接続ポリシー用の usNIC イーサネットアダプタプロファイルを指定します。この usNIC アダプタプロファイルは以前のバージョンの Cisco UCS Manager からリリース 2.2 にアップグレードするとデフォルトで作成されます。
ステップ 12	UCS-A /org/usnic-conn-policy* # commit-buffer	トランザクションをシステムの設定にコミットします。

次に、Cisco usNIC を作成してそのプロパティを指定する例を示します。

```
Server # scope org
Server /org # create usnic-conn-policy usnic1
Server /org/usnic-conn-policy* # set usnic-count 58
Server /org/usnic-conn-policy* # set adaptor-profile usNIC
Server /org/usnic-conn-policy* # commit-buffer
Server /org/usnic-conn-policy # top

Server # scope service-profile server 1/1
Server /org/service-profile # show vnic

vNIC:
Name Fabric ID Dynamic MAC Addr Virtualization Preference
-----
eth0 A 00:25:B5:00:00:A1 NONE
eth1 B 00:25:B5:00:00:A2 NONE
eth2 A 00:25:B5:00:00:A3 NONE
Server /org/service-profile # scope vnic eth0
Server /org/service-profile/vnic # set adapter-policy Linux
Server /org/service-profile/vnic # enter usnic-conn-policy-ref usnic1
Server /org/service-profile/vnic/usnic-conn-policy-ref* # commit-buffer
Server /org/service-profile/vnic/usnic-conn-policy-ref # exit
```

Cisco UCS Manager CLI を使用した usNIC の変更

はじめる前に

このタスクを実行するには、admin 権限を持つユーザとしてログインする必要があります。

手順の概要

1. UCS-A # **scope service-profile** *serverchassis-id/blade-id* or *rack_server-id*
2. UCS-A /org/service-profile # **show vnic**
3. UCS-A /org/service-profile # **scope vnic** *vnic name*
4. UCS-A /org/service-profile/vnic # **enter usnic-conn-policy-ref** *usnic connection policy reference name*
5. UCS-A /org/service-profile/vnic/usnic-conn-policy-ref* # **commit-buffer**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	UCS-A # scope service-profile serverchassis-id/blade-id or rack_server-id	指定されたシャーシ、ブレード、または UCS 管理ラック サーバ ID のサービス プロファイルを入力します。
ステップ 2	UCS-A /org/service-profile # show vnic	サーバ上で使用可能な vnic が表示されます。usnic vnic は Cisco UCS Manager リリース 2.2 にアップグレードするとデフォルトで使用できます。
ステップ 3	UCS-A /org/service-profile # scope vnicvnic name	指定された vNIC の vnic モードを開始します。
ステップ 4	UCS-A /org/service-profile/vnic # enter usnic-conn-policy-refusnic connection policy <i>reference name</i>	使用する vNIC 用の usnic 接続ポリシー参照を指定します。
ステップ 5	UCS-A /org/service-profile/vnic/usnic-conn-policy-ref* # commit-buffer	トランザクションをシステムの設定にコミットします。

次に、Cisco usNIC プロパティを変更する例を示します。

```
Server # scope service-profile server 1/1
Server /org/service-profile # show vnic

vNIC:
Name Fabric ID Dynamic MAC Addr Virtualization Preference
-----
eth0 A 00:25:B5:00:00:A1 SRIOV USNIC
eth1 B 00:25:B5:00:00:A2 NONE
eth2 A 00:25:B5:00:00:A3 NONE
Server /org/service-profile # scope vnic eth0
Server /org/service-profile/vnic # enter usnic-conn-policy-ref usnic2
Server /org/service-profile/vnic/usnic-conn-policy-ref* # commit-buffer
Server /org/service-profile/vnic/usnic-conn-policy-ref # exit
```

Cisco UCS Manager CLI を使用した usNIC の削除

はじめる前に

このタスクを実行するには、admin 権限を持つユーザとしてログインする必要があります。

手順の概要

1. UCS-A # **scope service-profile serverchassis-id/blade-id or rack_server-id**
2. UCS-A /org/service-profile # **show vnic**
3. UCS-A /org/service-profile # **scope vnicvnic name**
4. UCS-A /org/service-profile/vnic # **show usnic-conn-policy-refusnic connection policy reference name**
5. UCS-A /org/service-profile/vnic # **delete usnic-conn-policy-refusnic connection policy reference name**
6. UCS-A /org/service-profile/vnic/usnic-conn-policy-ref* # **commit-buffer**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	UCS-A # scope service-profile serverchassis-id/blade-id or rack_server-id	指定されたシャーシ、ブレード、または UCS 管理ラック サーバ ID のサービス プロファイルを入力します。
ステップ 2	UCS-A /org/service-profile # show vnic	サーバ上で使用可能な vNIC が表示されます。usNIC vNIC は Cisco UCS Manager リリース 2.2 にアップグレードするとデフォルトで使用できます。
ステップ 3	UCS-A /org/service-profile # scope vnicvnic name	指定された vNIC の vNIC モードを開始します。
ステップ 4	UCS-A /org/service-profile/vnic # show usnic-conn-policy-refusnic connection policy reference name	使用する vNIC 用の usNIC 接続ポリシー参照を指定します。
ステップ 5	UCS-A /org/service-profile/vnic # delete usnic-conn-policy-refusnic connection policy reference name	指定された usNIC 接続ポリシー参照を削除します。
ステップ 6	UCS-A /org/service-profile/vnic/usnic-conn-policy-ref* # commit-buffer	トランザクションをシステムの設定にコミットします。

次に、Cisco usNIC プロパティを変更する例を示します。

```
Server # scope service-profile server 1/1
Server /org/service-profile # show vnic

vNIC:
Name Fabric ID Dynamic MAC Addr Virtualization Preference
-----
eth0 A 00:25:B5:00:00:A1 SRIOV USNIC
eth1 B 00:25:B5:00:00:A2 NONE
eth2 A 00:25:B5:00:00:A3 NONE
Server /org/service-profile # scope vnic eth0
Server /org/service-profile/vnic # show usnic-conn-policy-ref

usNIC Connection Policy Reference:
usNIC Connection Policy Name
-----
usnic2
```

```
Server /org/service-profile/vnic # delete usnic-conn-policy-ref usnic2
Server /org/service-profile/vnic* # commit-buffer
Server /org/service-profile/vnic # exit
```

Cisco usNIC 用の Linux ソフトウェア パッケージのインストール

ここでは、UCS Drivers ISO バンドルに含まれているサポート対象の Linux オペレーティング システム ディストリビューションごとに usNIC フォルダの内容を列挙します。既知の問題やインストール手順に関するドキュメントが usNIC フォルダ内の README ファイルにも含まれています。

- **kmod-usnic_verbs-{version}.x86_64.rpm** : Cisco VIC SR-IOV イーサネット NIC の usNIC 機能用の Linux カーネル verbs ドライバ。
- **libdaplusnic_verbs-{version}.x86_64.rpm** : usNIC 用のユーザ スペース ライブラリ DAPL プラグイン。
- **openmpi-cisco-{version}.x86_64.rpm** : Cisco usNIC Open MPI : Cisco usNIC BTL MPI トランスポートを使用した Open MPI。
- **usnic_tools-{version}.x86_64.rpm** : usNIC 用のユーティリティ プログラム。
- **libfabric-cisco-{version}.x86_64.rpm** : 組み込みの Cisco usNIC トランスポートを備えた Libfabric パッケージ。
- **libusnic_verbs-{version}.x86_64.rpm** : libibverbs ライブラリに Cisco usNIC Linux デバイスをスキップさせるダミー ライブラリ (Cisco usNIC 機能は libibverbs ではなく、libfabric を介して公開されるため)。

はじめる前に

Cisco UCS Manager で Cisco usNIC プロパティが設定されていることを確認してください。プロパティの設定方法の詳細については、[Cisco usNIC の設定](#)を参照してください。

また、Cisco usNIC をインストールするホスト OS ディストリビューションに、サポート対象のバージョンの Cisco enic ドライバがインストールされていることを確認する必要があります。Cisco enic ドライバは Cisco VIC SR-IOV イーサネット NIC 用の Linux カーネル ネットワーキング ドライバです。

手順の概要

1. Linux ディストリビューション用の Cisco UCS ISO に含まれている enic ドライバの最新バージョンにアップグレードします。
2. Linux ディストリビューション用の Cisco UCS Drivers ISO から Cisco usNIC ソフトウェア パッケージをインストールします。
3. **# chkconfig rdma on**
4. サーバをリブートすると、インストールの変更が自動的に反映されます。

手順の詳細

- ステップ 1** Linux ディストリビューション用の Cisco UCS ISO に含まれている `enic` ドライバの最新バージョンにアップグレードします。
- ステップ 2** Linux ディストリビューション用の Cisco UCS Drivers ISO から Cisco usNIC ソフトウェア パッケージをインストールします。
- ステップ 3** `# chkconfig rdma on`
Linux RDMA サービスを有効にします。有効にしたら、システムのリブート後に RDMA サービスが自動的に開始されます。
- (注) RHEL 6.4 などの一部の Linux オペレーティング システム ディストリビューションでこの手順を実行しなければならない場合があります。
- ステップ 4** サーバをリブートすると、インストールの変更が自動的に反映されます。
- 重要** サーバをリブートしない場合は、手動でカーネル モジュールをロードし、システムによって正しいバージョンのドライバがロードされ、新しいメモリ ロック設定が適用されることを確認できます。モジュールのロード方法の詳細については、[Cisco usNIC のカーネルモジュールの手動のロード](#)を参照してください。

Linux Cisco usNIC ソフトウェア パッケージのソースコード

Cisco usNIC ソフトウェア パッケージのソース コードは Cisco UCS Drivers ISO で提供されます。ソース コードとバイナリ パッケージのインストールを混在させないことをお勧めします。

Cisco usNIC のカーネル モジュールの手動のロード

サーバをリブートしない場合、次の手順を使用して、Cisco usNIC カーネル モジュールを手動でロードできます。

はじめる前に

最新バージョンのドライバをロードする前に、既存バージョンのドライバが削除されていることを確認してください。これで、システムを正しく設定できます。

手順の概要

1. `# rmmod enic`
2. `# modprobe enic`
3. `# modprobe usnic_verbs`

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	# <code>rmmod enic</code>	既存の <code>enic</code> ドライバ モジュールをアンロードします。 (注) たとえば、ネットワークを使用して SSH 経由で OS にログインしていないことを確認します。そうしない場合、ネットワーク接続が完全に切断されている状態になる可能性があります。または、KVM を使用してサーバにログインして、この手順を実行することもできます。
ステップ 2	# <code>modprobe enic</code>	<code>enic</code> ドライバ モジュールをロードします。
ステップ 3	# <code>modprobe usnic_verbs</code>	<code>usnic_verbs</code> ドライバ モジュールをロードします。

Cisco usNIC シスコの Linux ソフトウェア パッケージのアップグレード

-
- ステップ 1 [Cisco usNIC 用の Linux ソフトウェア パッケージのアンインストール](#), (14 ページ) の手順に従って、usNIC ソフトウェア パッケージの以前のバージョンをアンインストールします。
- ステップ 2 [Cisco usNIC 用の Linux ソフトウェア パッケージのインストール](#), (12 ページ) の手順に従って、Linux ディストリビューション用の Cisco UCS Drivers ISO から usNIC ソフトウェア パッケージをインストールします。
-

Cisco usNIC 用の Linux ソフトウェア パッケージのアンインストール

手順の概要

1. 次の usNIC ソフトウェア パッケージをアンインストールします。
2. Cisco サーバをリブートします。

手順の詳細

-
- ステップ 1 次の usNIC ソフトウェア パッケージをアンインストールします。
- `usnic_tools`
 - `openmpi-cisco`

- libdaplusnic
- libusnic_verbs
- kmod-usnic_verbs
- libfabric-cisco-devel
- libfabric-cisco

ステップ 2 Cisco サーバをリブートします。

ユーザ環境への MPI の追加

MPI アプリケーションをコンパイルして起動するには、MPI 実装を各ユーザの環境に追加する必要があります。MPI 実装は一度に 1 つずつユーザの環境に追加することをお勧めします。

Cisco Open MPI の環境

Cisco Open MPI では、`openmpi-cisco` ソフトウェア パッケージが必要な環境変数の設定を支援するために 2 つのスクリプトをインストールします。1 つのスクリプトは Bourne シェル ユーザ用で、もう 1 つのスクリプトは C シェル ユーザ用です。

- `/opt/cisco/openmpi-vars.sh` (Bourne シェルおよび Bourne Again シェル ユーザ用)
- `/opt/cisco/openmpi-vars.csh` (C シェル ユーザ用)

ユーザのシェル起動/ログインシーケンス (非対話式シェルの場合) の一部として適切なスクリプトを調達する必要があります。

Intel® MPI ライブラリの環境

Intel® MPI ライブラリのドキュメントに記載されている手順に加えて、各ユーザの環境で追加の環境変数を設定して Cisco usNIC 機能を有効にする必要があります。必要な環境変数を設定するための 2 つのスクリプトが `libdaplusnic` ソフトウェア パッケージを通してインストールされます。1 つのスクリプトは Bourne シェル ユーザ用で、もう 1 つのスクリプトは C シェル ユーザ用です。

- `/opt/cisco/intelmpi-usnic-vars.sh` (Bourne シェルおよび Bourne Again シェル ユーザ用)
- `/opt/cisco/intelmpi-usnic-vars.csh` (C シェル ユーザ用)

ユーザのシェル起動/ログインシーケンスの一部として適切なスクリプトを調達する必要があります。

Intel® MPI ライブラリと usNIC を使用するには、ネットワーク上のフロー制御を有効に設定する必要があります。これは、IEEE 802.3x リンク レベルフロー制御と IEEE 802.1Qbb 優先順位ベースフロー制御（PFC）のどちらかにすることができます。この機能は、「no-drop」と呼ばれることもあります。フロー制御の有効化方法については、ネットワーク上のスイッチのコンフィギュレーションガイドを参照してください。フロー制御がネットワーク上で有効になっていない場合は、Intel® MPI ライブラリを使用するアプリケーションが正常に機能しますが、パフォーマンスが大幅に低下する可能性があります。

Intel® MPI ライブラリの開発では、MPI トラフィックはすべての Cisco usNIC ポートおよび Cisco UCS Manager のデフォルトの CoS 値 5 を持つ no-drop または platinum QoS システム クラスでフロー制御が有効になっている必要があります。「Cisco UCS Manager Network Management guide」で、「Quality of Service」に関するセクションを参照してください。

ユーザ環境への Libfabric の追加

Libfabric 用に、必要な環境変数を設定するための 2 つのスクリプトが libdaplusnic を通してインストールされます。1 つのスクリプトは Bourne シェルユーザ用で、もう 1 つのスクリプトは C シェルユーザ用です。

- /opt/cisco/libfabric-vars.sh
- /opt/cisco/libfabric-vars.csh

ユーザのシェル起動/ログインシーケンスの一部として適切なスクリプトを調達する必要があります。

Cisco UCS B シリーズ ブレード サーバ向け Cisco usNIC 取り付けの確認

Cisco usNIC に必要な Linux ドライバをインストールした後、Linux プロンプトで次の手順を実行して、インストールが正常に完了したことを確認してください。



(注) 次に、Linux オペレーティング システム ディストリビューション RHEL 6.5 上で確認された設定例を示します。

手順の概要

1. **usnic_verbs** カーネル モジュールが OS ドライバのインストール中にロードされたかどうかを検索して確認します。
2. Cisco usNIC 対応 NIC の設定を確認してください。
3. インストールされている RPM とそのバージョンを確認するには、**usnic_check** スクリプトを実行します。
4. Cisco usNIC ネットワーク パケットがクライアントとサーバ ホスト間で正常に送信されていることを確認します。
5. **ring_c** テストプログラムをダウンロード、コンパイル、および実行して、MPI トラフィックがクライアントとサーバのホスト間で正しく送信されていることを検証します。

手順の詳細

ステップ 1 **usnic_verbs** カーネル モジュールが OS ドライバのインストール中にロードされたかどうかを検索して確認します。

```
$ lsmod | grep usnic_verbs
```

lsmod | grep usnic_verbs コマンドを入力すると、次の詳細が表示されます。コンソールに一覧表示されているカーネル モジュールは、OS に現在ロードされているモジュールに基づいて異なる場合があります。

重要 **usnic_verbs** が出力に表示され、このカーネル モジュールがカーネル内にロードされ、アクティブになっていることが示されます。

```
usnic_verbs          73762  2
ib_core              74355  11
ib_ipoib,rdma_ucm,ib_ucm,ib_uverbs,ib_umad,rdma_cm,ib_cm,iw_cm,ib_sa,ib_mad,usnic_verbs
enic                 73723  1 usnic_verbs
```

ステップ 2 Cisco usNIC 対応 NIC の設定を確認してください。

```
$ /opt/cisco/usnic/bin/usd_devinfo
```

次のセクションは、**usd_devinfo** コマンドを実行すると表示される結果の簡単な例です。結果は、現在のインストール環境によって異なる場合があります。結果がコンソールに表示されたら、一覧表示されたポートのそれぞれの状態が **PORT_ACTIVE** と表示されていることを確認します。一覧表示されている各ポートのリンク状態がアップになっていることを確認します。

次に、Cisco UCS VIC アダプタ上で設定された 2 つのポート (**usnic_1** と **usnic_0**) の例を示します。1 つの Cisco usNIC 対応 vNIC だけを設定した場合は、**usnic_0** のみのリストが表示されます。

```
usnic_0:
Interface:          eth1
MAC Address:        00:25:b5:32:31:10
IP Address:         10.35.11.111
Netmask:            255.255.255.0
Prefix len:         24
MTU:                9000
Link State:         UP
Bandwidth:          10 Gb/s
Device ID:          UCSB-MLOM-40G-03 [VIC 1340] [0x012c]
Firmware:           4.1 (0.306)
VFs:                16
CQ per VF:         6
QP per VF:         6
```

```

        Interrupts per VF:      6
        Max CQ:                 96
        Max CQ Entries:        65535
        Max QP:                 96
        Max Send Credits:      4095
        Max Recv Credits:      4095
        Capabilities:
          CQ sharing:          yes
          PIO Sends:          yes
          CQ interrupts:      no
usnic_1:
        Interface:              eth2
        MAC Address:            00:25:b5:32:31:20
        IP Address:             10.35.12.111
        Netmask:                255.255.255.0
        Prefix len:             24
        MTU:                    9000
        Link State:             UP
        Bandwidth:              10 Gb/s
        Device ID:              UCSB-MLOM-40G-03 [VIC 1340] [0x012c]
        Firmware:               4.1(0.306)
        VFs:                    16
        CQ per VF:              6
        QP per VF:              6
        Interrupts per VF:      6
        Max CQ:                 96
        Max CQ Entries:        65535
        Max QP:                 96
        Max Send Credits:      4095
        Max Recv Credits:      4095
        Capabilities:
          CQ sharing:          yes
          PIO Sends:          yes
          CQ interrupts:      no

```

ステップ 3 インストールされている RPM とそのバージョンを確認するには、**usnic_check** スクリプトを実行します。

```
$ /opt/cisco/usnic/bin/usnic_check
```

エラーが OS ドライバのインストール中に発生している場合、警告が生成されます。

usnic_verbs モジュールのロードで失敗した場合は、次の簡単な例が生成された警告を示します。

```

$ /opt/cisco/usnic/bin/usnic_check
enic RPM version 2.3.0.18 installed
usnic_verbs RPM version 1.0.4.416.rhel6u5 installed
WARNING: usnic_verbs module not loaded
libdaplusnic RPM version 2.0.39cisco1.0.0.401 installed
libfabric RPM version 1.1.1cisco1.0.0.410.rhel6u5 installed
libusnic_verbs RPM version 2.0.0.406.rhel6u5 installed
Using /opt/cisco/openmpi/bin/mpi_info to check Open MPI info...
Open MPI version 1.10.1cisco1.0.0.410.rhel6u5 installed
WARNING: No usnic devices found
WARNING: No usnic devices found
3 warnings

```

ステップ 4 Cisco usNIC ネットワーク パケットがクライアントとサーバホスト間で正常に送信されていることを確認します。

a) サーバホストの Cisco usNIC に関連付けられているイーサネット インターフェイスの名前を判別します。

```
[server]$ /opt/cisco/usnic/bin/usnic_status
usnic_0: 0000:07:0.0, eth1, 00:25:b5:c4:b1:10, 58 VFs
  Per VF: 6 WQ, 6 RQ, 6 CQ, 6 INT
```

```
In use:
```

```
0 VFs, 0 QPs, 0 CQs
```

```
usnic_1: 0000:0c:0.0, eth2, 00:25:b5:c4:b1:20, 58 VFs
Per VF: 6 WQ, 6 RQ, 6 CQ, 6 INT
```

```
In use:
0 VFs, 0 QPs, 0 CQs
```

- b) イーサネット インターフェイスの IP アドレスを判別します。

```
[server]$ ip addr show dev eth1 | grep "inet[^6]"
inet 10.42.110.11/24 brd 50.42.110.255 scope global eth1
```

- c) サーバ ホストで **usd_pingpong** プログラムを実行します。

```
[server]$ /opt/cisco/usnic/bin/usd_pingpong
```

usd_pingpong プログラムで使用されるコマンドライン オプションの詳細については、**usd_pingpong --help** の出力を参照してください。

- d) サーバホスト上の Cisco usNIC に対応する IP アドレスを使用して、クライアントホストで **usd_pingpong** プログラムを実行します。

```
[client]# /opt/cisco/usnic/bin/usd_pingpong -h SERVER_IP_ADDRESS
```

次に、**usd_pingpong** プログラムを実行すると表示される結果の例を示します。

Server-side:

```
[server]$ /opt/cisco/usnic/bin/usd_pingpong -d usnic_0
open usnic_0 OK, IP=10.43.10.1
QP create OK, addr -h 10.43.10.1 -p 3333
Waiting for setup...
```

Client-side:

```
[client]# /opt/cisco/usnic/bin/usd_pingpong -h 10.43.10.1 -d usnic_0
open usnic_0 OK, IP=10.43.10.2
QP create OK, addr -h 10.43.10.2 -p 3333
sending params...
payload size=4, pkt_size=46
posted 63 RX buffers, size=64 (4)
100000 pkts, 4.953 us / HRTT
```

ステップ 5 **ring_c** テストプログラムをダウンロード、コンパイル、および実行して、MPI トラフィックがクライアントとサーバのホスト間で正しく送信されていることを検証します。

ring_c テストプログラムは、https://raw.githubusercontent.com/open-mpi/ompi-release/v1.8/examples/ring_c.c のリンクから取得できます。

次の例は、**wget** ユーティリティを使用して **ring_c** を取得、コンパイル、および実行する方法を示します。また、テストプログラムの取得および実行のその他の方法を使用できます。

(注) 環境内の単一の MPI 実装セットアップを使用して次のコマンドを実行します。

```
$ wget --no-check-certificate
https://raw.githubusercontent.com/open-mpi/ompi-release/v1.8/examples/ring_c.c
--2015-04-23 10:11:42--
https://raw.githubusercontent.com/open-mpi/ompi-release/v1.8/examples/ring_c.c
Resolving raw.githubusercontent.com... 199.27.74.133
Connecting to raw.githubusercontent.com|199.27.74.133|:443... connected.
WARNING: certificate common name \u201cwww.github.com\u201d doesn\u2019t match requested host name
\u201craw.githubusercontent.com\u201d.
HTTP request sent, awaiting response... 200 OK
Length: 2418 (2.4K) [text/plain]

$ mpicc ring_c.c -o ring_c
[no output]
```

```

$ mpiexec --host host1,host2 -n 4 ./ring_c
Process 0 sending 10 to 1, tag 201 (4 processes in ring)
Process 0 sent to 1
Process 0 decremented value: 9
Process 0 decremented value: 8
Process 0 decremented value: 7
Process 0 decremented value: 6
Process 0 decremented value: 5
Process 0 decremented value: 4
Process 0 decremented value: 3
Process 0 decremented value: 2
Process 0 decremented value: 1
Process 0 decremented value: 0
Process 0 exiting
Process 2 exiting
Process 1 exiting
Process 3 exiting ...

```

(注) 必要に応じて、環境内に別の MPI 実装をセットアップして、**mpicc** コマンドと **mpirun** コマンドを再実行し、Cisco usNIC 機能でその MPI 実装を確認します。

usd_pingpong プログラムと **ring_c** プログラムが正常に実行した場合は、Cisco usNIC 上で MPI アプリケーションを実行できるはずです。

トラブルシューティング情報

問題

usnic_check を使用してインストール済み RPM のリストを表示すると、次のことが発生します。

- 1 「**No usnic devices found**」などの警告。
- 2 「**usnic_verbs_xxxx does not match installed version**」などのバージョン mismatch エラー。

考えられる原因

インストールされているバージョンがこのエラーを引き起こしている可能性があります。

解決策

- 1 次のコマンドを使用して、インストールされているすべてのバージョンを一覧表示します。
rpm -qa|grep usnic_verbs
- 2 次のコマンドを使用してすべてのバージョンをアンインストールします。**rpm -e**
- 3 モジュールが削除されたことを確認します。
- 4 すべての RPM を再インストールします。

問題

usnic_pingpong を使用して Cisco usNIC パケットがクライアントとサーバ間で正しく送信されていることを確認すると、次のエラーが発生します。

- 1 「No such address or device」エラー。次の例を参照してください。

```
# /opt/cisco/usnic/bin/usd_pingpong
usd_open: No such device or address
```

- 2 サーバが「Waiting for setup...」の段階で停止しています。次の例を参照してください。

```
# /opt/cisco/usnic/bin/usd_pingpong -d usnic_0
open usnic_0 OK, IP=10.43.10.1
QP create OK, addr -h 10.43.10.1 -p 3333
Waiting for setup...
```

考えられる原因

- 1 Cisco usNIC 接続ポリシーが割り当てられていないか、「not set」になっているため、vNIC インターフェイスで設定されません。
- 2 サーバ側がクライアント側からパケットを受信しません。

解決策

- 1 有効な Cisco usNIC 接続ポリシーが usNIC 接続ポリシー内で設定され、サービス プロファイル内の vNIC に割り当てられていることを確認します。
- 2 サーバとクライアントの両方で Cisco usNIC デバイスの IP アドレスが正しく設定されていることを確認します。
- 3 クライアント ピンポンが Cisco usNIC デバイスの正しいサーバ IP アドレスにパケットを送信しようとしていることを確認します。

問題

mpirun を使用して Cisco usNIC トラフィックを処理すると、次のエラーが発生します。

MTU サイズ ミスマッチ エラー 次の例を参照してください。

```
Example:
$ mpirun --host node05,node06 -np 12 --mca btl usnic,sm,self --mca btl_usnic_if_include
usnic_1 IMB-MPI1 Sendrecv
```

```
The MTU does not match on local and remote hosts. All interfaces on
all hosts participating in an MPI job must be configured with the same
MTU. The usNIC interface listed below will not be used to communicate
with this remote host.
```

```
Local host:      node05
usNIC interface: usnic_1
Local MTU:       8958
Remote host:     node06
Remote MTU:      1458
```

考えられる原因

- 1 MTU サイズが適切な VLAN 上で正しく設定されていません。
- 2 MTU サイズが QoS 内で正しく設定されていません。

解決策

MTU サイズが VLAN と QoS 上で正しく設定されていることを確認します。

「[Configuring QoS System Classes with the LAN Uplinks Manager](#)」を参照してください。

問題

Cisco enic ドライバによって、次の Cisco enic 依存性エラーが発生します。

```
# rpm -ivh kmod-usnic_verbs-1.0.4.318.rhel6u5-1.x86_64.rpm
error: Failed dependencies:
        ksym(enic_api_devcmd_proxy_by_index) = 0x107cb661 is needed by
kmod-usnic_verbs-1.0.4.318.rhel6u5-1.x86_64
        ksym(vnic_dev_alloc_discover) = 0xfb7e4707 is needed by
kmod-usnic_verbs-1.0.4.318.rhel6u5-1.x86_64
        ksym(vnic_dev_get_pdev) = 0xae6ae5c9 is needed by
kmod-usnic_verbs-1.0.4.318.rhel6u5-1.x86_64
        ksym(vnic_dev_get_res) = 0xd910c86b is needed by
kmod-usnic_verbs-1.0.4.318.rhel6u5-1.x86_64
        ksym(vnic_dev_get_res_bar) = 0x31710a7e is needed by
kmod-usnic_verbs-1.0.4.318.rhel6u5-1.x86_64
        ksym(vnic_dev_get_res_bus_addr) = 0x7be7a062 is needed by
kmod-usnic_verbs-1.0.4.318.rhel6u5-1.x86_64
        ksym(vnic_dev_get_res_count) = 0x759e4b07 is needed by
kmod-usnic_verbs-1.0.4.318.rhel6u5-1.x86_64
        ksym(vnic_dev_get_res_type_len) = 0xd122f0a1 is needed by
kmod-usnic_verbs-1.0.4.318.rhel6u5-1.x86_64
        ksym(vnic_dev_unregister) = 0xd99602a1 is needed by
kmod-usnic_verbs-1.0.4.318.rhel6u5-1.x86_64
#
```

考えられる原因

- 1 enic ドライバが正しくインストールされていません。
- 2 enic ドライバがインストールされていません。

解決策

正しい enic ドライバがインストールされていることを確認します。また、次のことを確認します。

- 具体的に、次のことを確認します。enic ドライバおよび usnic_verbs ドライバは一致している必要があります。不一致があると、上記のバージョンエラーが発生する可能性があります。
- 具体的には、Cisco UCS ドライバ ISO に含まれる enic および usnic_verbs が互いに一致している必要があります。ある Cisco UCS ドライバ ISO の enic を使用し、別の Cisco UCS ドライバ ISO の usnic_verbs を使用した場合、上記のバージョンエラーが発生することになります。

問題

Intel IOMMU により次の警告が発生します。

```
# rpm -ivh kmod-usnic_verbs-1.0.4.318.rhel6u5-1.x86_64.rpm
Preparing... ##### [100%]
 1:kmod-usnic_verbs ##### [100%]
WARNING -
Intel IOMMU does not appear to be enabled - please add kernel parameter
intel_iommu=on to your boot configuration for USNIC driver to function.
#
```

考えられる原因

Intel IOMMU サポートが、Linux カーネルで有効になっていません。

解決策

Linux カーネルの Intel IOMMU ドライバを有効にします。

問題

DAT ユーザライブラリをインストールすると、次の障害が発生した依存関係によるエラーが生じます。

```
# rpm -ivh libdaplusnic-2.0.39cisco1.0.0.317-1.el6.x86_64.rpm
error: Failed dependencies:
        dapl is needed by libdaplusnic-2.0.39cisco1.0.0.317-1.el6.x86_64
#
```

考えられる原因

libdapl が DAT ライブラリをインストールせずにインストールされています。

解決策

DAT ライブラリをインストールします。

問題

usd_devinfo を使用して Cisco usNIC が有効化された VICS の構成を表示すると、コマンド出力が usNIC インターフェイスを一覧表示しません。

考えられる原因

RDMA サービスが有効になっていません。

解決策

次のコマンドを使用して RDMA サービスを有効にします。

```
# service rdma start
Or
# chkconfig rdma on
```

問題

openmpi-cisco パッケージをインストールすると、以下に示す依存関係エラーが生じます。

```
# rpm -ivh openmpi-cisco-1.10.1cisco1.0.0.424.rhel7u1-1.x86_64.rpm
error: Failed dependencies:
libfabric-cisco is needed by openmpi-cisco-1.10.1cisco1.0.0.424.rhel7u1-1.x86_64
libfabric.so.1()(64bit) is needed by openmpi-cisco-1.10.1cisco1.0.0.424.rhel7u1-1.x86_64
libfabric.so.1(FABRIC_1.0)(64bit) is needed by
openmpi-cisco-1.10.1cisco1.0.0.424.rhel7u1-1.x86_64
```

考えられる原因

libfabric-cisco パッケージがインストールされていないか、または存在しません。

解決策

openmpi-cisco パッケージをインストールする前に、libfabric-cisco パッケージをインストールします。

問題

libfabric-cisco パッケージを削除すると、次の障害が発生した依存関係エラーが生じます。

```
# rpm -e libfabric-cisco-1.1.0cisco1.0.0.385.rhel7u1-1.el7.x86_64
error: Failed dependencies:
libfabric-cisco is needed by (installed) openmpi-cisco-1.10.0cisco1.0.0.385.rhel7u1-1.x86_64
libfabric.so.1()(64bit) is needed by (installed)
openmpi-cisco-1.10.0cisco1.0.0.385.rhel7u1-1.x86_64
libfabric.so.1(FABRIC 1.0)(64bit) is needed by (installed)
openmpi-cisco-1.10.0cisco1.0.0.385.rhel7u1-1.x86_64
```

考えられる原因

libfabric rpm は削除できません。

解決策

libfabric パッケージを削除する前に openmpi-cisco パッケージを削除します。

