Red Hat Enterprise Linux 8

InfiniBand ネットワークおよび RDMA ネットワークの設定

Red Hat Enterprise Linux 8 で InfiniBand ネットワークおよび RDMA ネットワークを設定するためのガイド

Last Updated: 2020-09-07

Red Hat Enterprise Linux 8 InfiniBand ネットワークおよび RDMA ネットワークの設定

Red Hat Enterprise Linux 8 で InfiniBand ネットワークおよび RDMA ネットワークを設定するためのガイド

法律上の通知

概要

本ガイドでは、InfiniBand およびリモートダイレクトメモリーアクセス (RDMA) の概要と、InfiniBand ハードウェアの設定方法を説明します。また、InfiniBand 関連サービスの設定方法も説明します。

目次

RED HAT ドキュメントへのフィードバック (英語のみ)

ご意見ご要望をお聞かせください。ドキュメントの改善点はございませんか。改善点を報告する場合は、以下のように行います。

- 特定の文章に簡単なコメントを記入する場合は、以下の手順を行います。
    1. ドキュメントの表示が **Multi-page HTML** 形式になっていて、ドキュメントの右上端に **Feedback** ボタンがあることを確認してください。
    2. マウスカーソルで、コメントを追加する部分を強調表示します。
    3. そのテキストの下に表示される **Add Feedback** ポップアップをクリックします。
    4. 表示される手順に従ってください。
- より詳細なフィードバックを行う場合は、Bugzilla のチケットを作成します。
    1. Bugzilla の Web サイトにアクセスします。
    2. Component で **Documentation** を選択します。
    3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも記入してください。
    4. **Submit Bug** をクリックします。

第1章 INFINIBAND および RDMA について

InfiniBand は、以下の 2 つを指します。

- InfiniBand ネットワーク用の物理リンク層プロトコル
- リモートダイレクトメモリーアクセス (RDMA) テクノロジーの実装である InfiniBand Verbs API

RDMA は、どちらのコンピューターのオペレーティングシステムも必要とせずに、あるコンピューターから別のコンピューターのメモリーへのアクセスを提供します。このテクノロジーにより、CPU 使用量が低く、高スループットおよび低レイテンシーのネットワークが可能になります。

通常の IP データ転送では、あるマシンのアプリケーションが別のマシンのアプリケーションにデータを送信すると、受信側で次のことが起こります。

1. カーネルがデータを受信する必要がある。
2. カーネルが、データがアプリケーションに属するかどうかを判別する必要がある。
3. カーネルは、アプリケーションを起動する。
4. カーネルは、アプリケーションがカーネルへのシステムコールを実行するまで待機する。
5. アプリケーションは、データをカーネルの内部メモリー領域から、アプリケーションが提供するバッファーにコピーする。

このプロセスは、ホストアダプターがダイレクトメモリーアクセス (DMA) を使用する場合にはシステムのメインメモリーにほとんどのネットワークトラフィックをコピーするか、または少なくとも 2 回コピーされることを意味します。また、コンピューターは数多くのコンテキストスイッチを実行して、カーネルとアプリケーションコンテキストを切り替えます。どちらのコンテキストスイッチも、トラフィック速度が速くなると CPU 負荷が高くなり、他のタスクが遅くなる可能性があります。

RDMA 通信は、通常の IP 通信とは異なり、通信プロセスでのカーネルの介入を回避します。これにより、CPU のオーバーヘッドが軽減されます。RDMA プロトコルにより、ホストアダプターは、ネットワークからパケットを受信するタイミング、受信するアプリケーション、およびアプリケーションのメモリー領域内でパケットを保存する場所を認識できます。パケットをカーネルに送信して処理される代わりに、InfiniBand でユーザーアプリケーションのメモリーにコピーする代わりに、ホストアダプターはパケットの内容をアプリケーションのバッファーに直接保存します。このプロセスでは、RDMA を使用する前に、個別の API、InfiniBand Verbs API、およびアプリケーションがこの API に対応している必要があります。

Red Hat Enterprise Linux 8 は、InfiniBand ハードウェアと InfiniBand Verbs API の両方に対応しています。また、Red Hat Enterprise Linux は、InfiniBand 以外のハードウェアで InfiniBand Verbs API を使用できるようにする以下のテクノロジーに対応しています。

- iWARP (Internet Wide Area RDMA Protocol) - IP ネットワーク上で RDMA を実装するネットワークプロトコル。
- RoCE (RDMA over Converged Ethernet) (IBoE (InfiniBand over Ethernet) とも呼ばれます) - RDMA over Ethernet ネットワークを実装するネットワークプロトコル。

関連情報

- RoCE のソフトウェア実装の設定に関する詳細は、*2章RoCE の設定*を参照してください。

第2章 ROCE の設定

本セクションでは、RoCE (RDMA over Converged Ethernet) の背景情報、デフォルトの RoCE バージョンを変更する方法、およびソフトウェア RoCE アダプターの設定方法を説明します。

RoCE ハードウェアを提供するベンダー (Mellanox、Broadcom、QLogic など) が異なることに注意してください。

2.1. ROCE プロトコルバージョンの概要

RoCE は、イーサネット経由のリモートダイレクトメモリーアクセス (RDMA) を有効にするネットワークプロトコルです。

以下は、RoCE のさまざまなバージョンです。

RoCE v1

RoCE バージョン 1 プロトコルは、同じイーサネットブロードキャストドメインの任意のホスト間の通信を可能にするイーサネットタイプ **0x8915** を持つイーサネットリンク層プロトコルです。
デフォルトでは、Mellanox ConnectX-3 ネットワークアダプターを使用する場合、Red Hat Enterprise Linux は RDMA Connection Manager (RDMA_CM) に RoCE v1 を使用します。

RoCE v2

RoCE バージョン 2 プロトコルは、UDP over IPv4 または UDP over IPv6 プロトコルのいずれかにあります。UDP 宛先ポート番号 4791 は RoCE v2 用に予約されています。
デフォルトでは、Mellanox ConnectX-3 Pro、ConnectX-4 Lx、または ConnectX-5 のネットワークアダプターを使用する場合、Red Hat Enterprise Linux は RDMA_CM に RoCE v2 を使用しますが、ハードウェアは RoCE v1 と RoCE v2 の両方に対応します。

RDMA_CM は、データを転送するためにクライアントとサーバーとの間に信頼できる接続を設定します。RDMA_CM は、接続を確立するために RDMA トランスポートに依存しないインターフェースを提供します。通信は特定の RDMA デバイスを使用し、データ転送はメッセージベースです。

重要

クライアントでの RoCE v2 の使用と、サーバーでの RoCE v1 の使用には対応していません。この場合は、サーバーとクライアントの両方が RoCE v1 で通信するように設定します。

関連情報

- 「デフォルトの RoCE バージョンを一時的に変更」

2.2. デフォルトの ROCE バージョンを一時的に変更

クライアントで RoCE v2 プロトコルを使用し、サーバーの RoCE v1 には対応していません。サーバーのハードウェアが RoCE v1 にのみ対応している場合は、RoCE v1 を使用してサーバーと通信するようにクライアントを設定します。本セクションでは、Mellanox ConnectX-5 Infiniband デバイスの **mlx5_0** ドライバーを使用するクライアントで RoCE v1 を強制する方法を説明します。本セクションで説明している変更は、ホストを再起動するまでの一時的なものです。

前提条件

- クライアントが、デフォルトでは RoCE v2 プロトコルを使用する InfiniBand デバイスを使用し

ている。

- サーバーの InfiniBand デバイスが RoCE v1 のみに対応している。

手順

1. **/sys/kernel/config/rdma_cm/mlx5_0/** ディレクトリーを作成します。

```
# mkdir /sys/kernel/config/rdma_cm/mlx5_0/
```

2. デフォルトの RoCE モードを表示します。たとえば、ポート 1 のモードを表示するには、次のコマンドを実行します。

```
# cat /sys/kernel/config/rdma_cm/mlx5_0/ports/1/default_roce_mode

  RoCE v2
```

3. デフォルトの RoCE モードをバージョン 1 に変更します。

```
# echo "IB/RoCE v1" > /sys/kernel/config/rdma_cm/mlx5_0/ports/1/default_roce_mode
```

2.3. SOFT-ROCE の設定

Soft-RoCE は、イーサネット経由のリモートダイレクトメモリーアクセス (RDMA) のソフトウェア実装で、RXE とも呼ばれます。本セクションでは、Soft-RoCE を設定する方法を説明します。

RoCE ホストチャンネルアダプター (HCA) のないホストで Soft-RoCE を使用します。

前提条件

- システムにイーサネットアダプターがインストールされている。

手順

1. **libibverbs** パッケージ、**libibverbs-utils** パッケージ、および **infiniband-diags** パッケージをインストールします。

```
# yum install libibverbs libibverbs-utils infiniband-diags
```

2. **rdma_rxe** カーネルモジュールを読み込み、現在の設定を表示します。

```
# rxe_cfg start
  Name    Link  Driver      Speed  NMTU  IPv4_addr       RDEV  RMTU
  enp7s0  yes   virtio_net         1500
```

3. 新しい RXE デバイスを追加します。たとえば、**enp7s0** イーサネットデバイスを RXE デバイスとして追加するには、次のコマンドを実行します。

```
# rxe_cfg add enp7s0
```

4. RXE デバイスステータスを表示します。

```
# rxe_cfg status
  Name    Link  Driver      Speed  NMTU  IPv4_addr       RDEV  RMTU
  enp7s0  yes   virtio_net        1500                  rxe0  1024  (3)
```

RDEV 列で、**enp7s0** が **rxe0** デバイスにマッピングされていることを確認します。

5. 必要に応じて、システムで利用可能な RDMA デバイスの一覧を表示します。

```
# ibv_devices
    device                 node GUID
    ------              ----------------
    rxe0                505400fffed5e0fb
```

または、**ibstat** ユーティリティーを使用して詳細なステータスを表示します。

```
# ibstat rxe0
CA 'rxe0'
 CA type:
 Number of ports: 1
 Firmware version:
 Hardware version:
 Node GUID: 0x505400fffed5e0fb
 System image GUID: 0x0000000000000000
 Port 1:
  State: Active
  Physical state: LinkUp
  Rate: 100
  Base lid: 0
  LMC: 0
  SM lid: 0
  Capability mask: 0x00890000
  Port GUID: 0x505400fffed5e0fb
  Link layer: Ethernet
```

第3章 コア RDMA サブシステムの設定

本セクションでは、**rdma** サービスを設定し、ユーザーがシステムで固定 (ピン留め) できるメモリーの量を増やす方法を説明します。

3.1. RDMA サービスの設定

rdma サービスは、カーネルの RDMA スタックを管理します。Red Hat Enterprise Linux が、InfiniBand デバイス、iWARP デバイス、または RoCE デバイスを検出すると、**udev** デバイスマネージャーが **systemd** に **rdma** サービスを開始するよう指示します。

手順

1. **/etc/rdma/rdma.conf** ファイルを編集し、有効にするモジュールの変数を **yes** に設定します。以下は、Red Hat Enterprise Linux 8 のデフォルトの **/etc/rdma/rdma.conf** です。

```
# Load IPoIB
IPOIB_LOAD=yes
# Load SRP (SCSI Remote Protocol initiator support) module
SRP_LOAD=yes
# Load SRPT (SCSI Remote Protocol target support) module
SRPT_LOAD=yes
# Load iSER (iSCSI over RDMA initiator support) module
ISER_LOAD=yes
# Load iSERT (iSCSI over RDMA target support) module
ISERT_LOAD=yes
# Load RDS (Reliable Datagram Service) network protocol
RDS_LOAD=no
# Load NFSoRDMA client transport module
XPRTRDMA_LOAD=yes
# Load NFSoRDMA server transport module
SVCRDMA_LOAD=no
# Load Tech Preview device driver modules
TECH_PREVIEW_LOAD=no
```

2. **rdma** サービスを再起動します。

```
# systemctl restart rdma
```

3.2. IPOIB デバイスの名前変更

デフォルトでは、カーネルには IPoIB (IP over InfiniBand) デバイス (**ib0**、**ib1**など) という名前が付けられます。競合を回避するために、Red Hat は **udev** デバイスマネージャーでルールを作成し、**mlx4_ib0** などの永続的で意味のある名前を作成することが推奨されます。

前提条件

- InfiniBand デバイスがホストにインストールされている。

手順

1. デバイスのハードウェアアドレスを表示します。たとえば、**ib0** という名前のデバイスのアドレスを表示するには、次のコマンドを実行します。

```
# ip link show ib0
8: ib0: >BROADCAST,MULTICAST,UP,LOWER_UP< mtu 65520 qdisc pfifo_fast state UP
mode DEFAULT qlen 256
    link/infiniband 80:00:02:00:fe:80:00:00:00:00:00:00:00:00:02:c9:03:00:31:78:f2 brd
00:ff:ff:ff:ff:12:40:1b:ff:ff:00:00:00:00:00:00:ff:ff:ff:ff
```

次の例で **udev** ルールを作成するには、例で太字で示されているアドレスの最後の 8 バイトが必要です。

2. **/etc/udev/rules.d/70-persistent-ipoib.rules** ファイルを編集し、**ACTION** ルールを追加します。たとえば、**00:02:c9:03:00:31:78:f2** ハードウェアアドレスでデバイスの名前を変更するルールを **mlx4_ib0** に設定するには、以下の行を追加します。

```
ACTION=="add", SUBSYSTEM=="net", DRIVERS=="?*", ATTR{type}=="32",
ATTR{address}=="?*00:02:c9:03:00:31:78:f2", NAME="mlx4_ib0"
```

3. ホストを再起動します。

```
# reboot
```

関連情報

- **udev** ルールの詳細は、man ページの **udev(7)** を参照してください。
- **udev** ルールで、ハードウェアアドレスの最初の 12 バイトが使用されていない詳細な理由は、「IPoIB ハードウェアアドレスについて」を参照してください。

3.3. システムの固定 (ピン留め) にユーザーが使用できるメモリー量の増加

RDMA 操作には、物理メモリーのピン留めが必要です。これは、カーネルがメモリーをスワップ領域に書き込むことができないことを意味します。ユーザーがメモリーを過剰にピン留めすると、システムのメモリーが不足し、カーネルがプロセスを終了してより多くのメモリーを解放できます。このため、メモリーのピン留めは特権付きの操作になります。

root 以外のユーザーが大規模な RDMA アプリケーションを実行する場合は、システムでこれらのユーザーがピン留めするメモリー容量を増やす必要があります。本セクションでは、**rdma** グループのメモリー使用量を無制限に設定する方法を説明します。

手順

- **root** ユーザーで、以下の内容で **/etc/security/limits.d/rdma.conf** ファイルを作成します。

```
@rdma    soft    memlock     unlimited
@rdma    hard    memlock     unlimited
```

検証手順

1. **/etc/security/limits.d/rdma.conf** ファイルを編集したら **rdma** グループのメンバーとしてログインします。
   Red Hat Enterprise Linux は、ユーザーのログイン時に、更新された **ulimit** の設定を適用することに注意してください。
2. **ulimit -l** コマンドを使用して制限を表示します。

```
$ ulimit -l
unlimited
```

コマンドが **unlimited** を返すと、ユーザーはメモリーのピン留めを無制限にできます。

関連情報

- システムリソースの制限の詳細は、man ページの **limits.conf(5)** を参照してください。

第4章 INFINIBAND サブネットマネージャーの設定

すべての InfiniBand ネットワークでは、ネットワークが機能するために、サブネットマネージャーが実行している必要があります。これは、2 台のマシンがスイッチなしで直接接続されている場合にも当てはまります。

複数のサブネットマネージャーを使用することもできます。その場合は、サブネットマネージャーの 1 つがマスターとして機能し、マスターサブネットマネージャーが失敗した場合に引き継げるように、別のサブネットマネージャーがスレーブとして機能します。

ほとんどの InfiniBand スイッチには、埋め込みサブネットマネージャーが含まれます。ただし、最新のサブネットマネージャーが必要な場合や、制御が必要な場合は、Red Hat Enterprise Linux が提供する **OpenSM** サブネットマネージャーを使用します。

4.1. OPENSM サブネットマネージャーのインストール

本セクションでは、OpenSM サブネットマネージャーをインストールする方法を説明します。

手順

1. **opensm** パッケージをインストールします。

```
# yum install opensm
```

2. デフォルトインストールがお使いの環境に一致しない場合に OpenSM を設定します。InfiniBand ポートが 1 つ だけインストールされている場合は、ホストがマスターサブネットマネージャーとして機能する必要があり、カスタムの変更は必要ありません。デフォルト設定は変更せずに動作します。

3. **opensm** サービスを有効にして開始します。

```
# systemctl enable --now opensm
```

関連情報

- **opensm** サービスのコマンドラインオプションの一覧、およびパーティション構成、サービスの品質 (QoS)、およびその他の高度なトピックの詳細は、man ページの **opensm(8)** を参照してください。

4.2. 簡単な方法での OPENSM の設定

本セクションでは、カスタマイズした設定が必要ない場合に OpenSM を設定する方法を説明します。

前提条件

- 1 つ以上の InfiniBand ポートがサーバーにインストールされている。

手順

1. **ibstat** ユーティリティーを使用してポートの GUID を取得します。

```
# ibstat -d device_name
CA 'mlx4_0'
```

```
CA type: MT4099
Number of ports: 2
Firmware version: 2.42.5000
Hardware version: 1
Node GUID: 0xf4521403007be130
System image GUID: 0xf4521403007be133
Port 1:
  State: Active
  Physical state: LinkUp
  Rate: 56
  Base lid: 3
  LMC: 0
  SM lid: 1
  Capability mask: 0x02594868
  Port GUID: 0xf4521403007be131
  Link layer: InfiniBand
Port 2:
  State: Down
  Physical state: Disabled
  Rate: 10
  Base lid: 0
  LMC: 0
  SM lid: 0
  Capability mask: 0x04010000
  Port GUID: 0xf65214fffe7be132
  Link layer: Ethernet
```

注記

一部の InfiniBand アダプターでは、ノード、システム、およびポートに、同じ GUID を使用します。

2. **/etc/sysconfig/opensm** ファイルを編集し、**GUIDS** パラメーターで GUID を設定します。

```
GUIDS="GUID_1 GUID_2"
```

3. 必要に応じて、サブネットで複数のサブネットマネージャーが利用可能な場合には、**PRIORITY** パラメーターを設定します。以下に例を示します。

```
PRIORITY=15
```

関連情報

- **/etc/sysconfig/opensm** で設定できるパラメーターの詳細は、そのファイルのドキュメントを参照してください。

4.3. OPENSM.CONF ファイルを編集して OPENSM の設定

本セクションでは、**/etc/rdma/opensm.conf** ファイルを編集して OpenSM を設定する方法を説明します。利用可能な InfiniBand ポートが 1 つだけの場合は、この方法を使用して OpenSM 設定をカスタマイズできます。

前提条件

- サーバーに InfiniBand ポートが 1 つだけインストールされている。

手順

1. **/etc/rdma/opensm.conf** ファイルを編集し、お使いの環境に合わせて設定をカスタマイズします。
2. **opensm** サービスを再起動します。

```
# systemctl restart opensm
```

関連情報

- 更新した **opensm** パッケージをインストールすると、**yum** ユーティリティーが新しい OpenSM 設定ファイルを **/etc/rdma/opensm.conf.rpmnew** として保存します。このファイルを、カスタマイズした **/etc/rdma/opensm.conf** ファイルと比較して、手動で変更を加えます。

4.4. 複数の OPENSM インスタンスの設定

本セクションでは、OpenSM の複数のインスタンスを設定する方法を説明します。

前提条件

- 1 つ以上の InfiniBand ポートがサーバーにインストールされている。

手順

1. 必要に応じて、**/etc/rdma/opensm.conf** ファイルを **/etc/rdma/opensm.conf.orig** ファイルにコピーします。

```
# cp /etc/rdma/opensm.conf /etc/rdma/opensm.conf.orig
```

更新した **opensm** パッケージをインストールすると、**yum** ユーティリティーが **/etc/rdma/opensm.conf** を上書きします。この手順で作成したコピーで、以前のファイルと新しいファイルを比較して変更を特定し、インスタンス固有の **opensm.conf** ファイルで手動で取り入れることができます。

2. **/etc/rdma/opensm.conf** ファイルのコピーを作成します。

```
# cp /etc/rdma/opensm.conf /etc/rdma/opensm.conf.1
```

作成するインスタンスごとに、設定ファイルのコピーに一意の連続した番号を追加します。

3. 前の手順で作成したコピーを編集し、お使いの環境に合わせて、インスタンスの設定をカスタマイズします。たとえば、**guid** パラメーター、**subnet_prefix** パラメーター、および **logdir** パラメーターを設定します。
4. 必要に応じて、このサブネット専用の一意の名前で **partitions.conf** ファイルを作成し、**opensm.conf** ファイルの対応するコピーの **partition_config_file** パラメーターでそのファイルを参照します。
5. 作成するインスタンスごとに、前の手順を繰り返します。
6. **opensm** サービスを開始します。

```
# systemctl start opensm
```

opensm サービスは、**/etc/rdma/** ディレクトリー内の各 **opensm.conf.*** ファイルで一意のインスタンスを自動的に開始します。複数の **opensm.conf.*** ファイルが存在する場合、サービスは **/etc/sysconfig/opensm** ファイルおよびベースファイル **/etc/rdma/opensm.conf** の設定を無視します。

関連情報

- 更新した **opensm** パッケージをインストールすると、**yum** ユーティリティーが新しい OpenSM 設定ファイルを **/etc/rdma/opensm.conf.rpmnew** として保存します。このファイルを、カスタマイズした **/etc/rdma/opensm.conf.*** ファイルと比較して、手動で変更を加えます。

4.5. パーティション設定の作成

本セクションでは、OpenSM 向けに InfiniBand パーティション設定を作成する方法を説明します。パーティションを使用すると、管理者はイーサネット VLAN と同じように、InfiniBand にサブネットを作成できます。

重要

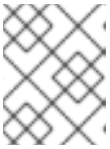
40 Gbps などの特定の速度でパーティションを定義する場合は、このパーティション内のすべてのホストが少なくともこの速度を対応している必要があります。ホストが速度要件を満たさない場合は、パーティションに参加できません。したがって、パーティションの速度を、パーティションに参加することが許可されているホストが対応する最低速度に設定します。

前提条件

- 1 つ以上の InfiniBand ポートがサーバーにインストールされている。

手順

1. **/etc/rdma/partitions.conf** ファイルを編集し、パーティションを設定します。



注記

すべてのファブリックには **0x7fff** パーティションが含まれ、すべてのスイッチとすべてのホストがそのファブリックに属する必要があります。

たとえば、次の内容をファイルに追加して、減速した 10 Gbps でデフォルトパーティション **0x7fff** を作成し、40 Gbps の速度でパーティション **0x0002** を作成します。

```
# For reference:
# IPv4 IANA reserved multicast addresses:
#   http://www.iana.org/assignments/multicast-addresses/multicast-addresses.txt
# IPv6 IANA reserved multicast addresses:
#   http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xml
#
# mtu =
#   1 = 256
#   2 = 512
```

```
#   3 = 1024
#   4 = 2048
#   5 = 4096
#
# rate =
#   2 =   2.5 GBit/s
#   3 =  10   GBit/s
#   4 =  30   GBit/s
#   5 =   5   GBit/s
#   6 =  20   GBit/s
#   7 =  40   GBit/s
#   8 =  60   GBit/s
#   9 =  80   GBit/s
#   10 = 120   GBit/s

Default=0x7fff, rate=3, mtu=4, scope=2, defmember=full:
    ALL, ALL_SWITCHES=full;
Default=0x7fff, ipoib, rate=3, mtu=4, scope=2:
    mgid=ff12:401b::ffff:ffff   # IPv4 Broadcast address
    mgid=ff12:401b::1           # IPv4 All Hosts group
    mgid=ff12:401b::2           # IPv4 All Routers group
    mgid=ff12:401b::16          # IPv4 IGMP group
    mgid=ff12:401b::fb          # IPv4 mDNS group
    mgid=ff12:401b::fc          # IPv4 Multicast Link Local Name Resolution group
    mgid=ff12:401b::101         # IPv4 NTP group
    mgid=ff12:401b::202         # IPv4 Sun RPC
    mgid=ff12:601b::1           # IPv6 All Hosts group
    mgid=ff12:601b::2           # IPv6 All Routers group
    mgid=ff12:601b::16          # IPv6 MLDv2-capable Routers group
    mgid=ff12:601b::fb          # IPv6 mDNS group
    mgid=ff12:601b::101         # IPv6 NTP group
    mgid=ff12:601b::202         # IPv6 Sun RPC group
    mgid=ff12:601b::1:3         # IPv6 Multicast Link Local Name Resolution group
    ALL=full, ALL_SWITCHES=full;

ib0_2=0x0002, rate=7, mtu=4, scope=2, defmember=full:
        ALL, ALL_SWITCHES=full;
ib0_2=0x0002, ipoib, rate=7, mtu=4, scope=2:
    mgid=ff12:401b::ffff:ffff   # IPv4 Broadcast address
    mgid=ff12:401b::1           # IPv4 All Hosts group
    mgid=ff12:401b::2           # IPv4 All Routers group
    mgid=ff12:401b::16          # IPv4 IGMP group
    mgid=ff12:401b::fb          # IPv4 mDNS group
    mgid=ff12:401b::fc          # IPv4 Multicast Link Local Name Resolution group
    mgid=ff12:401b::101         # IPv4 NTP group
    mgid=ff12:401b::202         # IPv4 Sun RPC
    mgid=ff12:601b::1           # IPv6 All Hosts group
    mgid=ff12:601b::2           # IPv6 All Routers group
    mgid=ff12:601b::16          # IPv6 MLDv2-capable Routers group
    mgid=ff12:601b::fb          # IPv6 mDNS group
    mgid=ff12:601b::101         # IPv6 NTP group
    mgid=ff12:601b::202         # IPv6 Sun RPC group
    mgid=ff12:601b::1:3         # IPv6 Multicast Link Local Name Resolution group
    ALL=full, ALL_SWITCHES=full;
```

第5章 IPOIB の設定

デフォルトでは、InfiniBand は通信にインターネットプロトコル (IP) を使用しません。ただし、IPoIB (IP over InfiniBand) は、InfiniBand リモートダイレクトメモリーアクセス (RDMA) ネットワーク上に IP ネットワークエミュレーション層を提供します。これにより、既存の変更されていないアプリケーションが InfiniBand ネットワーク上でデータを送信できますが、アプリケーションが RDMA をネイティブで使用する場合よりもパフォーマンスが低くなります。

注記

iWARP (Internet Wide Area RDMA Protocol) および RoCE ネットワークはすでに IP ベースです。したがって、IWARP デバイスまたは RoCE デバイスに IPoIB デバイスを作成することはできません。

5.1. IPOIB 通信モード

IPoIB デバイスは、**Datagram** モードまたは **Connected** モードで設定できます。違いは、通信の反対側で IPoIB 層がマシンで開こうとするキューペアのタイプです。

- **Datagram** モードでは、システムは信頼できない非接続キューのペアを開きます。
  このモードは、InfiniBand リンク層の MTU (Maximum Transmission Unit) を超えるパッケージには対応していません。IPoIB レイヤーは、送信される IP パケット上に 4 バイトの IPoIB ヘッダーを追加します。これにより、IPoIB MTU は InfiniBand リンク層の MTU よりも 4 バイト小さくなければなりません。一般的な InfiniBand リンク層の MTU は 2048 であるため、**Datagram** モードの共通の IPoIB デバイスの MTU は 2044 です。

- **Connected** モードでは、システムは信頼できる接続されたキューペアを開きます。
  このモードでは、InfiniBand リンク層の MTU を超えるメッセージを許可し、ホストアダプターはパケットのセグメンテーションを処理し、再構築します。その結果、**Connected** モードで InfiniBand アダプターが送信できる IPoIB メッセージのサイズに制限はありません。ただし、IP パケットのサイズは、**size** フィールドと TCP/IP ヘッダーにより制限されます。このため、**Connected** モードの IPoIB MTU は最大 **65520** バイトです。

  Connected モードではパフォーマンスが向上しますが、より多くのカーネルメモリーを消費します。

システムが **Connected** モードを使用するように設定されている場合、InfiniBand スイッチおよびファブリックは **Connected** モードでマルチキャストトラフィックを通過できないため、システムは **Datagram** モードでマルチキャストトラフィックを送信します。また、**Connected** モードで設定されていないホストと通信すると、システムは **Datagram** モードに戻ります。

マルチキャストデータをインターフェースの最大 MTU に送信するアプリケーションを実行している間は、インターフェースを **Datagram** モードで設定するか、パケット送信サイズを、datagram サイズのパケットに適合するサイズに制限するようにアプリケーションを設定する必要があります。

5.2. IPOIB ハードウェアアドレスについて

IPoIB デバイスには、以下の部分で構成される 20 バイトのハードウェアアドレスがあります。

- 最初の 4 バイトはフラグとキューのペア番号です。

- 次の 8 バイトはサブネットの接頭辞です。
  デフォルトのサブネットの接頭辞は **0xfe:80:00:00:00:00:00:00** です。デバイスがサブネットマネージャーに接続すると、デバイスはこの接頭辞を、サブネットマネージャーで設定されたものに合わせて変更します。

- 最後の 8 バイトは、IPoIB デバイスが接続している InfiniBand ポートのグローバル一意識別子 (GUID) です。

注記

最初の 12 バイトは変更できるため、**udev** デバイスマネージャールールで使用しないでください。

関連情報

- IPoIB デバイスの名前変更の詳細は、「IPoIB デバイスの名前変更」を参照してください。

5.3. NMCLI コマンドを使用した IPOIB 接続の設定

この手順では、**nmcli** コマンドを使用して IPoIB 接続を設定する方法を説明します。

前提条件

- サーバーに InfiniBand デバイスがインストールされ、対応するカーネルモジュールが読み込まれている。

手順

1. InfiniBand 接続を作成します。たとえば、**Connected** トランスポートモードで **mlx4_ib0** インターフェースを使用し、最大 MTU の **65520** バイトを使用する接続を作成するには、次のコマンドを実行します。

```
# nmcli connection add type infiniband con-name mlx4_ib0 ifname mlx4_ib0 transport-mode Connected mtu 65520
```

2. 必要に応じて、**P_Key** インターフェースを設定します。たとえば、**mlx4_ib0** 接続の **P_Key** インターフェースとして **0x8002** を設定するには、次のコマンドを実行します。

```
# nmcli connection modify mlx4_ib0 infiniband.p-key 0x8002
```

3. IPv4 設定を構成します。たとえば、静的 IPv4 アドレス、ネットワークマスク、デフォルトゲートウェイ、および **mlx4_ib0** の DNS サーバーを設定するには、次のコマンドを実行します。

```
# nmcli connection modify mlx4_ib0 ipv4.addresses '192.0.2.1/24'
# nmcli connection modify mlx4_ib0 ipv4.gateway '192.0.2.254'
# nmcli connection modify mlx4_ib0 ipv4.dns '192.0.2.253'
# nmcli connection modify mlx4_ib0 ipv4.method manual
```

4. IPv6 設定を構成します。たとえば、静的 IPv6 アドレス、ネットワークマスク、デフォルトゲートウェイ、および **mlx4_ib0** 接続の DNS サーバーを設定するには、次のコマンドを実行します。

```
# nmcli connection modify mlx4_ib0 ipv6.addresses '2001:db8:1::1/32'
# nmcli connection modify mlx4_ib0 ipv6.gateway '2001:db8:1::fffe'
# nmcli connection modify mlx4_ib0 ipv6.dns '2001:db8:1::fffd'
# nmcli connection modify mlx4_ib0 ipv6.method manual
```

5. 接続をアクティベートします。たとえば、**mlx4_ib0** 接続を有効にするには、次のコマンドを実行します。

```
# nmcli connection up mlx4_ib0
```

5.4. NM-CONNECTION-EDITOR を使用した IPOIB 接続の設定

この手順では、**nm-connection-editor** アプリケーションを使用して IPoIB 接続を設定する方法を説明します。

前提条件

- サーバーに InfiniBand デバイスがインストールされており、対応するカーネルモジュールが読み込まれている。
- **nm-connection-editor** パッケージがインストールされている。

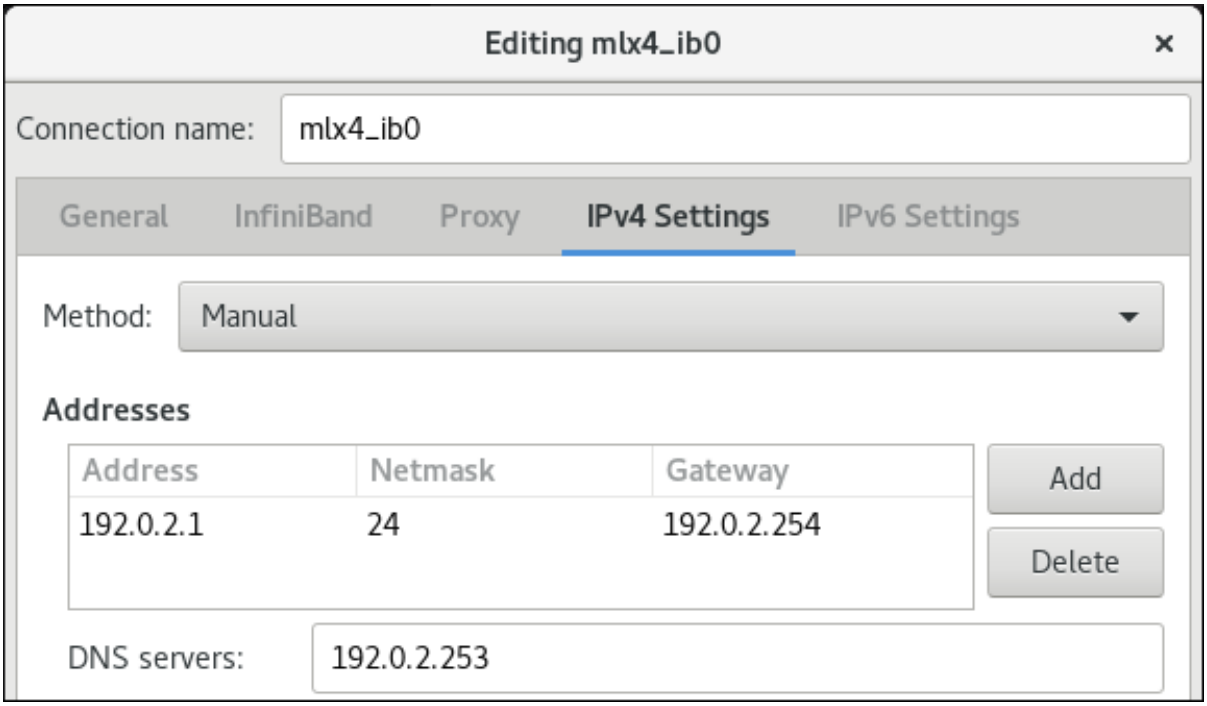
手順

1. 端末を開き、次のコマンドを入力します。

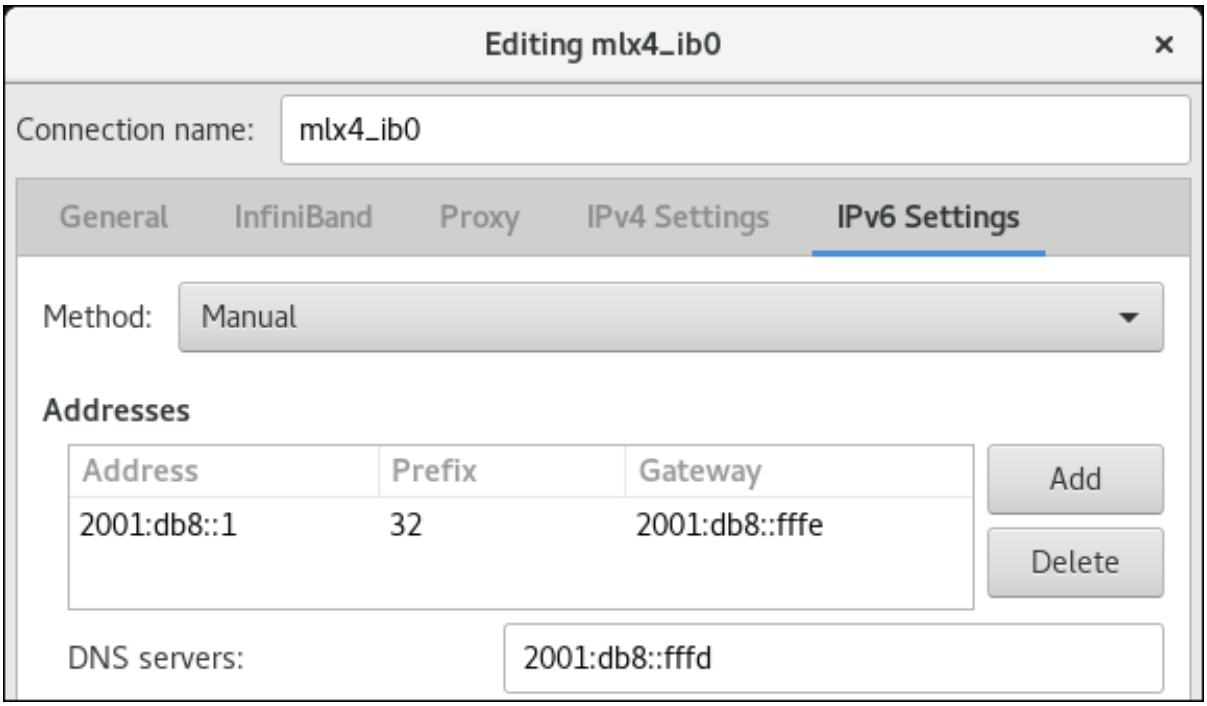
```
$ nm-connection-editor
```

2. **+** ボタンをクリックして、新しい接続を追加します。
3. **InfiniBand** 接続タイプを選択し、**Create** をクリックします。
4. **InfiniBand** タブで以下を行います。
   a. 必要に応じて、接続名を変更します。
   b. トランスポートモードを選択します。
   c. デバイスを選択します。
   d. 必要に応じて、MTU を設定します。

5. **IPv4 Settings** タブで、IPv4 設定を構成します。たとえば、静的な IPv4 アドレス、ネットワークマスク、デフォルトゲートウェイ、および DNS サーバーを設定します。



6. **IPv6 設定** タブで、IPv6 設定を構成します。たとえば、静的な IPv6 アドレス、ネットワークマスク、デフォルトゲートウェイ、および DNS サーバーを設定します。



7. **保存** をクリックして、チーム接続を保存します。

8. **nm-connection-editor** を閉じます。

9. 必要に応じて、**P_Key** インターフェースを設定します。**nm-connection-editor** ではこの設定を利用できないため、このパラメーターをコマンドラインで設定する必要があります。
たとえば、**mlx4_ib0** 接続の **P_Key** インターフェースとして **0x8002** を設定するには、次のコマンドを実行します。

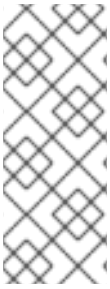
```
# nmcli connection modify mlx4_ib0 infiniband.p-key 0x8002
```

第6章 INFINIBAND ネットワークのテスト

本セクションでは、InfiniBand ネットワークをテストする手順を説明します。

6.1. 初期の INFINIBAND RDMA 操作のテスト

本セクションでは、InfiniBand リモートダイレクトメモリーアクセス (RDMA) 操作をテストする方法を説明します。



注記

このセクションは、InfiniBand デバイスにのみ適用されます。IP ベースの iWARP デバイスまたは RoCE/IBoE デバイスを使用する場合は、以下を参照してください。

- 「ping ユーティリティーを使用した IPoIB のテスト」
- 「IPoIB の設定後に qperf を使用した RDMA ネットワークのテスト」

前提条件

- RDMA が設定されている。
- **libibverbs-utils** パッケージおよび **infiniband-diags** パッケージがインストールされている。

手順

1. 利用可能な InfiniBand デバイスの一覧を表示します。

```
# ibv_devices
    device                 node GUID
    ------              ----------------
    mlx4_0              0002c903003178f0
    mlx4_1              f4521403007bcba0
```

2. 特定の InfiniBand デバイスの情報を表示します。たとえば、**mlx4_1** デバイスの情報を表示するには、次のコマンドを実行します。

```
# ibv_devinfo -d mlx4_1
hca_id: mlx4_1
    transport:                  InfiniBand (0)
    fw_ver:                     2.30.8000
    node_guid:                  f452:1403:007b:cba0
    sys_image_guid:             f452:1403:007b:cba3
    vendor_id:                  0x02c9
    vendor_part_id:             4099
    hw_ver:                     0x0
    board_id:                   MT_1090120019
    phys_port_cnt:              2
        port:   1
            state:              PORT_ACTIVE (4)
            max_mtu:            4096 (5)
            active_mtu:         2048 (4)
            sm_lid:             2
            port_lid:           2
```

```
            port_lmc:          0x01
            link_layer:        InfiniBand

        port:   2
            state:             PORT_ACTIVE (4)
            max_mtu:           4096 (5)
            active_mtu:        4096 (5)
            sm_lid:            0
            port_lid:          0
            port_lmc:          0x00
            link_layer:        Ethernet
```

3. InfiniBand デバイスの基本的なステータスを表示します。たとえば、**mlx4_1** デバイスのステータスを表示するには、次のコマンドを実行します。

```
# ibstat mlx4_1
CA 'mlx4_1'
    CA type: MT4099
    Number of ports: 2
    Firmware version: 2.30.8000
    Hardware version: 0
    Node GUID: 0xf4521403007bcba0
    System image GUID: 0xf4521403007bcba3
    Port 1:
        State: Active
        Physical state: LinkUp
        Rate: 56
        Base lid: 2
        LMC: 1
        SM lid: 2
        Capability mask: 0x0251486a
        Port GUID: 0xf4521403007bcba1
        Link layer: InfiniBand
    Port 2:
        State: Active
        Physical state: LinkUp
        Rate: 40
        Base lid: 0
        LMC: 0
        SM lid: 0
        Capability mask: 0x04010000
        Port GUID: 0xf65214fffe7bcba2
        Link layer: Ethernet
```

4. **ibping** ユーティリティーを使用して、InfiniBand を使用してクライアントからサーバーに ping します。

    a. サーバーとして機能するホストで、サーバーモードで **ibping** を起動します。

    ```
    # ibping -S -C mlx4_1 -P 1
    ```

    このコマンドは、以下のようなパラメーターを使用します。

    - **-S** - サーバーモードを有効にします。
    - **-C InfiniBand_CA_name** - 使用する CA 名を設定します。

- **-P port_number** - InfiniBand が複数のポートを提供する場合は、使用するポート番号を設定します。

b. クライアントとして動作するホストで、以下のように **ibping** を使用します。

```
# ibping -c 50 -C mlx4_0 -P 1 -L 2
```

- **-c number** - これらの数のパケットをサーバーに送信します。
- **-C InfiniBand_CA_name** - 使用する CA 名を設定します。
- **-P port_number** - InfiniBand が複数のポートを提供する場合は、使用するポート番号を設定します。
- **-L port_LID** - 使用するローカル識別子 (LID) を設定します。

関連情報

- **ibping** パラメーターの詳細は、man ページの **ibping(8)** を参照してください。

6.2. PING ユーティリティーを使用した IPOIB のテスト

IPoIB を設定したら、**ping** ユーティリティーを使用して ICMP パケットを送信して IPoIB 接続をテストします。

前提条件

- 2 台の RDMA ホストは、同じ InfiniBand ファブリックで RDMA ポートに接続されている。
- 両方のホストの IPoIB インターフェースは、同じサブネット内の IP アドレスで設定されている。

手順

1. **ping** ユーティリティーを使用して ICMP パケットをリモートホストの InfiniBand アダプターに送信します。

```
# ping -c5 192.0.2.1
```

このコマンドは、ICMP パケットを IP アドレス **192.0.2.1** に送信します。

6.3. IPOIB の設定後に QPERF を使用した RDMA ネットワークのテスト

この手順では、**qperf** ユーティリティーを使用して、InfiniBand アダプター設定を表示し、ホスト間の帯域幅とレイテンシーを測定する方法の例を説明します。

前提条件

- **qperf** パッケージが両方のホストにインストールされている。
- IPoIB が両方のホストに設定されている。

手順

1. サーバーとして機能するオプションを指定せずに、いずれかのホストで **qperf** を起動します。

```
# qperf
```

2. クライアントで以下のコマンドを使用します。コマンドは、クライアントの **mlx4_0** ホストチャネルアダプターのポート **1** を使用して、サーバーの InfiniBand アダプターに割り当てられた IP アドレス **192.0.2.1** に接続します。

   a. 設定を表示するには、以下を入力します。

```
qperf -v -i mlx4_0:1 192.0.2.1 conf
------------------------
conf:
    loc_node   =  rdma-dev-01.lab.bos.redhat.com
    loc_cpu    =  12 Cores: Mixed CPUs
    loc_os     =  Linux 4.18.0-187.el8.x86_64
    loc_qperf  =  0.4.11
    rem_node   =  rdma-dev-00.lab.bos.redhat.com
    rem_cpu    =  12 Cores: Mixed CPUs
    rem_os     =  Linux 4.18.0-187.el8.x86_64
    rem_qperf  =  0.4.11
------------------------
```

   b. Reliable Connection (RC) ストリーミングの双方向帯域幅を表示するには、以下を入力します。

```
# qperf -v -i mlx4_0:1 192.0.2.1 rc_bi_bw
------------------------
rc_bi_bw:
    bw              =  10.7 GB/sec
    msg_rate        =   163 K/sec
    loc_id          =  mlx4_0
    rem_id          =  mlx4_0:1
    loc_cpus_used   =    65 % cpus
    rem_cpus_used   =    62 % cpus
------------------------
```

   c. RC ストリーミングの一方向帯域幅を表示するには、以下を入力します。

```
# qperf -v -i mlx4_0:1 192.0.2.1 rc_bw
------------------------
rc_bw:
    bw              =  6.19 GB/sec
    msg_rate        =  94.4 K/sec
    loc_id          =  mlx4_0
    rem_id          =  mlx4_0:1
    send_cost       =  63.5 ms/GB
    recv_cost       =    63 ms/GB
    send_cpus_used  =  39.5 % cpus
    recv_cpus_used  =    39 % cpus
------------------------
```

関連情報

- **qperf** の詳細は、man ページの **qperf(1)** を参照してください。