Red Hat Enterprise Linux 8

配置 InfiniBand 和 RDMA 网络

在 Red Hat Enterprise Linux 8 中配置 InfiniBand 和 RDMA 网络的指南

Last Updated: 2021-06-07

Red Hat Enterprise Linux 8 配置 InfiniBand 和 RDMA 网络

在 Red Hat Enterprise Linux 8 中配置 InfiniBand 和 RDMA 网络的指南

法律通告

摘要

本文档描述了 InfiniBand 和远程直接访问(RDMA)是什么以及如何配置 InfiniBand 硬件。另外，本文档解释了如何配置与 InfiniBand 相关的服务。

目录

使开源包含更多

红帽承诺替换我们的代码、文档和网页属性中存在问题的语言。我们从这四个术语开始： master、slave、blacklist 和 whitelist。这些更改将在即将发行的几个发行本中逐渐实施。如需了解更多详细信息，请参阅 CTO Chris Wright 信息。

对红帽文档提供反馈

我们感谢您对文档提供反馈信息。请让我们了解如何改进文档。要做到这一点：

- 关于特定内容的简单评论：
  1. 请确定您使用 *Multi-page HTML* 格式查看文档。另外，确定 **Feedback** 按钮出现在文档页的右上方。
  2. 用鼠标指针高亮显示您想评论的文本部分。
  3. 点在高亮文本上弹出的 **Add Feedback**。
  4. 按照显示的步骤操作。
- 要提交更复杂的反馈，请创建一个 Bugzilla ticket：
  1. 进入 Bugzilla 网站。
  2. 在 Component 中选择 **Documentation**。
  3. 在 **Description** 中输入您要提供的信息。包括文档相关部分的链接。
  4. 点 **Submit Bug**。

第 1 章 了解 INFINIBAND 和 RDMA

InfiniBand 代表两个不同的因素：

- InfiniBand 网络的物理链路协议
- InfiniBand Verbs API，这是 RDMA（remote direct memory access）技术的一个实现

RDMA 可在不涉及计算机操作系统的情况下，从一个计算机访问另一台计算机的内存。此技术启用了高吞吐量和低延迟联网，且 CPU 使用率较低。

在典型的 IP 数据传输中，当机器中的某个应用程序向另一台机器上的应用程序发送数据时，在接收层时会出现以下情况：

1. 内核必须接收数据。
2. 内核必须确定该数据是否属于该应用程序。
3. 内核唤醒应用程序。
4. 内核会等待应用程序执行系统调用到内核。
5. 应用程序将内核本身的内部内存空间数据复制到应用程序提供的缓冲中。

这个过程意味着,如果主机适配器使用直接内存访问(DMA),或者至少两次,则大多数网络流量都会在系统主内存间复制。另外，计算机执行很多上下文开关以在内核和应用程序上下文间进行切换。这些上下文切换都可能造成高流量率的 CPU 负载，并可能造成其他任务的性能下降。

RDMA 通讯会绕过内核在沟通过程中的干预，这和普通的 IP 通讯不同这可减少 CPU 开销。RDMA 协议让主机适配器知道数据包何时来自网络，应用程序应该接收它,并在应用程序的内存空间中保存数据包。对于 InfiniBand，主机适配器不将数据包发送到内核，然后将其复制到用户应用程序的内存中，而是,主机适配器将数据包的内容直接放置在应用程序的缓冲中。此过程需要单独的 API、InfiniBand Verbs API，应用程序必须支持这个 API 才能使用 RDMA。

Red Hat Enterprise Linux 8 支持 InfiniBand 硬件和 InfiniBand Verbs API。另外，Red Hat Enterprise Linux 支持以下技术，以便在非 InfiniBand 硬件中使用 InfiniBand Verbs API:

- Internet Wide Area RDMA Protocol(iWARP):通过 IP 网络实施 RDMA 的网络协议。
- RDMA over Converged Ethernet(RoCE),也称为 InfiniBand over Ethernet(IBoE): 通过以太网实现 RDMA 的网络协议。

其它资源

- 有关设置 RoCE 软件实施的详情,请参考 第 2 章 *配置 RoCE*。

第 2 章 配置 ROCE

本节介绍使用 Converged Ethernet(RoCE)的 RDMA 的后台信息,以及如何更改默认 RoCE 版本以及如何配置软件 RoCE 适配器。

请注意，有不同的厂商，比如 Mellanox、Broadcom 和 QLogic 都提供 RoCE 硬件。

2.1. ROCE 协议版本概述

RoCE 是一个网络协议,它可通过以太网启用远程直接访问(RDMA)。

以下是不同的 RoCE 版本：

RoCE v1

RoCE 版本 1 协议是一个以太网链路层协议,它使用以太网类型 **0x8915**,它允许同一以太网广播域中的任何两个主机相互通信。
默认情况下，在使用 Mellanox ConnectX-3 网络适配器时，Red Hat Enterprise Linux 使用 RoCE v1 作为 RDMA 连接管理器（RDMA_CM）。

RoCE v2

RoCE 版本 2 协议在 IPv4 或 IPv6 协议的 UDP 上存在。RoCE v2 保留 UDP 目标端口号 4791。
默认情况下，在使用 Mellanox ConnectX-3 Pro、ConnectX-4 Lx 或 ConnectX-5 网络适配器时，Red Hat Enterprise Linux 将 RoCE v2 用于 RDMA_CM，但硬件支持 RoCE v1 和 RoCE v2。

RDMA_CM 设置客户端和服务器之间用来传输数据的可靠连接。RDMA_CM 为建立连接提供了一个与 RDMA 传输相关的接口。该通信使用特定的 RDMA 设备,数据传输是基于消息的。

重要

在客户端使用 RoCE v2，在服务器使用 RoCE v1 不被支持。在这种情况下，将服务器和客户端都配置为通过 RoCE v1 进行通信。

其它资源

- 如需更多信息, 请参阅参数更改默认 RoCE 版本。

2.2. 临时更改默认 ROCE 版本

在客户端使用 RoCE v2 协议,在服务器中使用 RoCE v1 不被支持。如果您的服务器中硬件只支持 RoCE v1，请将您的客户端配置为使用 RoCE v1 与服务器通信。这部分论述了如何在使用 Mellanox ConnectX-5 Infiniband 设备的客户端中强制 RoCE v1。**mlx5_0**请注意，本节中描述的更改只在重启主机前临时进行。

先决条件

- 客户端默认使用 RoCE v2 协议的 InfiniBand 设备。
- 服务器中的 InfiniBand 设备只支持 RoCE v1。

流程

1. 创建 **/sys/kernel/config/rdma_cm/*mlx5_0*/** 目录：

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

2. 显示默认 RoCE 模式。例如：显示端口 1 的模式：

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

  RoCE v2
```

3. 将默认 RoCE 模式改为版本 1:

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

2.3. 配置 SOFT-ROCE

Soft-RoCE 是 RDMA over Ethernet 的一个软件实现，它也称为 RXE。这部分论述了如何配置 Soft-RoCE。

在没有 RoCE 主机频道适配器(HCA)的主机上使用 Soft-RoCE。

先决条件

- 在系统中安装了一个以太网适配器。

流程

1. 安装 **libibverbs**、**libibverbs-utils** 和 **infiniband-diags** 软件包：

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

2. 载入 **rdma_rxe** 内核模块并显示当前的配置：

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

3. 添加一个新的 RXE 设备。例如：要将 **enp7s0** 以太网设备添加为 RXE 设备，请输入：

```
# rxe_cfg add enp7s0
```

4. 显示 RXE 设备状态：

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

在 **RDEV** 列中，您会看到该设备 **enp7s0** 已映射到 **rxe0** 设备。

5. 可选：列出系统中可用的 RDMA 设备：

```
# ibv_devices
    device                 node GUID
    ------              ----------------
```

```
    rxe0            505400fffed5e0fb
```

或者，使用 **ibstat** 工具显示详细状态：

```
# 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 配置核心 RDMA 子系统

这部分论述了如何配置 **rdma** 服务并增加用户允许在系统中固定的内存量。

3.1. 配置 RDMA 服务

rdma 服务在内核中管理 RDMA 堆栈。如果 Red Hat Enterprise Linux 检测到 InfiniBand、iWARP 或 RoCE 设备，则 **udev** 设备管理器会指示 **systemd** 启动 **rdma** 服务。

流程

1. 编辑 **/etc/rdma/rdma.conf** 文件，并设置要启用的模块的变量为 **yes**。以下是 **/etc/rdma/rdma.conf** Red Hat Enterprise Linux 8 中的默认设置：

```
# 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. 重启 **rdma** 服务：

```
# systemctl restart rdma
```

3.2. 重命名 IPOIB 设备

默认情况下，内核命名 IP over InfiniBand (IPoIB) 设备，例如 **ib0**、**ib1** 等。为避免冲突，红帽建议在 **udev** 设备管理器中创建一个规则来创建持久且有意义的名称，例如 **mlx4_ib0**。

先决条件

- 在主机上安装了 InfiniBand 设备。

流程

1. 显示该设备的硬件地址。例如：要显示名为 **ib0** 的设备的地址，请输入：

```
# 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:00:00:00:00:00:00:00:00:ff:ff:ff:ff
```

在下一步中创建 **udev** 规则需要最后的 8 字节地址（在示例中以粗体标记）。

2. 编辑 **/etc/udev/rules.d/70-persistent-ipoib.rules** 文件，并附加一个 **ACTION** 规则。例如：要配置将设备重命名为 **mlx4_ib0** 的规则,请附加以下行：**00:02:c9:03:00:31:78:f2**

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

3. 重启主机：

```
# reboot
```

其它资源

- 有关 **udev** 规则的详情,请查看 **udev(7)** man page。
- 详情请查看 **udev** 规则中没有使用硬件地址的前 12 字节的原因。请参阅 了解 IPoIB 硬件地址

3.3. 增加用户允许在系统中固定的内存量

RDMA 操作需要固定物理内存。这意味着内核不允许把内存写入到 swap 空间中。如果用户固定太多内存，系统会耗尽内存，并且内核会终止进程来释放更多内存。因此，内存固定是一个特权操作。

如果非 root 用户运行大型 RDMA 应用程序，则可能需要增加这些用户可在系统中的内存量。这部分论述了如何为 **rdma** 组群配置无限内存。

流程

- 以 **root** 用户身份,使用以下内容创建 **/etc/security/limits.d/rdma.conf** 该文件：

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

验证步骤

1. 编辑 **/etc/security/limits.d/rdma.conf** 文件后作为 **rdma** 组的成员登录。
请注意，当用户登录时，Red Hat Enterprise Linux 会应用更新的 **ulimit** 设置。

2. 使用 **ulimit -l** 命令显示限制：

```
$ ulimit -l
unlimited
```

如果命令返回 **unlimited**，用户可以获得无限数量的内存。

其它资源

- 有关限制系统资源的详情请参考 **limits.conf(5)** man page。

第 4 章 配置 INFINIBAND 子网管理器

所有 InfiniBand 网络都必须运行子网管理器才能正常工作。即使两台机器没有使用交换机直接进行连接，也是如此。

有可能有一个以上的子网管理器。在那种情况下，当主子网管理器出现故障时，另外一个作为从网管理器的系统会接管。

大多数 InfiniBand 交换机都包含一个嵌入式子网管理器。然而，如果您需要一个更新的子网管理器，或者您需要更多控制，请使用 Red Hat Enterprise Linux 提供的 **OpenSM** 子网管理器。

4.1. 安装 OPENSM 子网管理器

本节论述了如何安装 OpenSM 子网管理器。

流程

1. 安装 **opensm** 软件包：

```
# yum install opensm
```

2. 如果默认安装与您的环境不匹配，请配置 OpenSM。
   如果只安装一个 InfiniBand 端口,则主机应充当主子网管理器,且无需自定义更改。默认配置可在没有任何修改的情况下正常工作。

3. 启用并启动 **opensm** 服务：

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

其它资源

- 有关 **opensm** 服务的命令行选项列表,以及分区配置、服务质量(QoS)和其他高级主题的其他描述,请查看 **opensm(8)** man page。

4.2. 使用简单方法配置 OPENSM

这部分论述了如何在不需要自定义设置时配置 OpenSM。

先决条件

- 在服务器中安装一个或多个 InfiniBand 端口。

流程

1. 使用 **ibstat** 实用程序获取端口的 GUID：

```
# 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
```

注意

有些 InfiniBand 适配器在节点、系统和端口中使用相同的 GUID。

2. 编辑 **/etc/sysconfig/opensm** 文件并在 **GUIDS** 参数中设置 GUID：

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

3. 另外，如果您的子网中有多个子网管理器，也可以设置 **PRIORITY** 参数。例如：

```
PRIORITY=15
```

其它资源

- 有关您可以在 **/etc/sysconfig/opensm** 中设置的参数的附加信息，请参考该文件中的文档。

4.3. 通过编辑 OPENSM.CONF 文件配置 OPENSM

这部分论述了如何通过编辑 **/etc/rdma/opensm.conf** 文件配置 OpenSM。如果只有一个 InfiniBand 端口可用，则使用此方法自定义 OpenSM 配置。

先决条件

- 服务器上只安装一个 InfiniBand 端口。

流程

1. 编辑 **/etc/rdma/opensm.conf** 文件并自定义设置以匹配您的环境。

2. 重启 **opensm** 服务：

```
# systemctl restart opensm
```

其它资源

- 当您安装更新的 **opensm** 软件包时，该 **yum** 工具会将新的 OpenSM 配置文件保存为 **/etc/rdma/opensm.conf.rpmnew**。将这个文件与您自定义的 **/etc/rdma/opensm.conf** 文件进行比较，并手动纳入这些更改。

4.4. 配置多个 OPENSM 实例

这部分论述了如何设置多个 OpenSM 实例。

先决条件

- 在服务器中安装一个或多个 InfiniBand 端口。

流程

1. （可选）将 **/etc/rdma/opensm.conf** 文件复制到 **/etc/rdma/opensm.conf.orig** 文件：

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

   当您安装更新的 **opensm** 软件包时，**yum** 工具会覆盖 **/etc/rdma/opensm.conf**。在这个步骤中生成的副本时，您可以把前一个文件和新文件进行比较，以识别更改并将其手动合并到特定实例 **opensm.conf** 文件中。

2. 创建 **/etc/rdma/opensm.conf** 文件的副本：

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

   对于您创建的每个实例,在配置文件的副本中附加一个唯一连续数字。

3. 编辑您在上一步中创建的副本，并自定义实例的设置以匹配您的环境。例如，设置 **guid**、**subnet_prefix** 和 **logdir** 参数。

4. 另外，还可在该子网 **partitions.conf** 中生成具有唯一名称的文件，并在该文件对应的副本中的 **partition_config_file** 参数中引用 **opensm.conf** 文件。

5. 对您要创建的每个实例重复前面的步骤。

6. 启动 **opensm** 服务：

   ```
   # systemctl start opensm
   ```

   opensm 服务自动为 **/etc/rdma/** 目录中的每个 **opensm.conf.*** 文件启动一个唯一的实例。如果存在多个 **opensm.conf.*** 文件，该服务会忽略 **/etc/sysconfig/opensm** 文件中以及基本文件中的 **/etc/rdma/opensm.conf** 设置。

其它资源

- 当您安装更新的 **opensm** 软件包时，该 **yum** 工具会将新的 OpenSM 配置文件保存为 **/etc/rdma/opensm.conf.rpmnew**。将这个文件与您自定义的 **/etc/rdma/opensm.conf.*** 文件进行比较并手动纳入这些更改。

4.5. 创建分区配置

这部分论述了如何为 OpenSM 创建 InfiniBand 分区配置。分区使管理员能够在 InfiniBand 上创建与以太网 VLAN 类似的子网。

重要

如果您使用特定速度定义分区，比如 40 Gbps，这个分区中的所有主机必须至少支持这个速度。如果主机没有满足速度要求，就无法加入该分区。因此,将分区速度设置为任何有权限加入分区的主机支持的最低速度。

先决条件

- 在服务器中安装一个或多个 InfiniBand 端口。

流程

1. 编辑 **/etc/rdma/partitions.conf** 文件并配置分区。

注意

所有光纤必须包含 **0x7fff** 分区，所有交换机和所有主机都必须属于那个光纤。

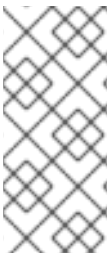
例如：在该文件中添加以下内容以便创建 **0x7fff** 默认分区使用较慢的速度 10 Gbps，另一个分区的速度 **0x0002** 为 40 Gbps：

```
# 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 配置 IPOIB

默认情况下，InfiniBand 不使用 IP 进行通信。但是,IP over InfiniBand(IPoIB)在 InfiniBand 远程直接访问(RDMA)网络之上提供了一个 IP 网络模拟层。这允许现有的未修改应用程序通过 InfiniBand 网络传输数据,但性能低于应用程序原生使用 RDMA 时。



注意

互联网 Wide Area RDMA 协议(iWARP)和 RoCE 网络已经基于 IP。因此，您不能在 IWARP 或 RoCE 设备之上创建 IPoIB 设备。

从 RHEL 8 上的 ConnectX-4 及更高版本开始的 Mellanox 设备默认使用 Enhanced IPoIB 模式（仅用于数据图）。这些设备不支持连接的模式。

5.1. IPOIB 通讯模式

您可以在 **Datagram** 或 **Connected** 模式下配置 IPoIB 设备。不同之处在，IPoIB 层试图使用什么类型的队列对在通信的另一端的机器中打开：

- 在 **Datagram** 模式下，系统会打开一个不可靠、断开连接的队列对。
  这个模式不支持大于 InfiniBand link-layer 的 Maximum Transmission unit(MTU)的软件包。IPoIB 层在传输的 IP 数据包之上添加了一个 4 字节 IPoIB 标头。因此，IPoIB MTU 需要比 InfiniBand link-layer MTU 小 4 字节。因为 2048 是一个常见的 InfiniBand link-layer MTU, **Datagram** 模式中的通用 IPoIB 设备 MTU 是 2044。

- 在 **Connected** 模式下，系统会打开一个可靠、连接的队列对。
  这个模式允许大于 InfiniBand link-layer MTU 的消息,主机适配器处理数据包片段并重新编译。因此，InfiniBand 适配器在 **Connected** 模式下发送的 IPoIB 信息大小没有限制。但是，IP 数据包会受 **size** 字段和 TCP/IP 标头的限制。因此， **Connected** 模式中的 IPoIB MTU 是 **65520** 字节的最大值。

  Connected 模式性能更高，但是消耗更多内核内存。

如果系统被配置为使用 **Connected** 模式，它仍然会以 **Datagram** 模式发送多播流量，因为 InfiniBand 交换机和光纤无法在 **Connected** 模式中传递多播流量。另外，当与没有在 **Datagram** 模式中配置的主机进行通信时，系统会返回 **Connected** 模式。

在运行应用程序时，将多播数据发送到接口的最大 MTU 时，您必须将接口配置为 **Datagram** 模式，或者将应用程序配置为以数据包大小数据包的大小封顶数据包发送的大小。

5.2. 了解 IPOIB 硬件地址

ipoib 设备有 20 个字节硬件地址，它由以下部分组成：

- 前 4 字节是标志和队列对号。

- 下一个 8 字节是子网前缀。
  默认子网前缀是 **0xfe:80:00:00:00:00:00:00**。设备连接到子网管理器后，设备会修改这个前缀使其与子网管理器中配置的匹配。

- 最后的 8 字节是 IPoIB 设备附加到 InfiniBand 端口的全局唯一识别符(GUID)。

注意

因为前 12 字节可以改变，所以不要在 **udev** 设备管理器规则中使用它们。

其它资源

- 有关重命名 IPoIB 设备的详情请参考 第 3.2 节 “重命名 IPoIB 设备”。

5.3. 使用 NMCLI 命令配置 IPOIB 连接

这个步骤描述了如何使用 **nmcli** 命令配置 IPoIB 连接。

先决条件

- 在服务器中安装 InfiniBand 设备，并载入相应的内核模块。

流程

1. 创建 InfiniBand 连接。例如，要创建一个连接，在 **Connected** 传输模式中使用 **mlx4_ib0** 接口和最大 MTU **65520** 字节数，请输入：

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

2. 可选：设置 **P_Key** 接口。例如，要将 **0x8002** 设置为 **mlx4_ib0** 连接的 **P_Key** 接口，请输入：

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

3. 配置 IPv4 设置。例如：要设置静态 IPv4 地址、网络掩码、默认网关和 **mlx4_ib0** 连接的 DNS 服务器，请输入：

```
# 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. 配置 IPv6 设置。例如，要设置静态 IPv6 地址、网络掩码、默认网关和 **mlx4_ib0** 连接的 DNS 服务器，请输入：

```
# 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. 激活连接。例如，激活 **mlx4_ib0** 连接：

```
# nmcli connection up mlx4_ib0
```

5.4. 使用 NM-CONNECTION-EDITOR 配置 IPOIB 连接

这个步骤描述了如何使用 **nm-connection-editor** 应用程序配置 IPoIB 连接。

先决条件

先决条件

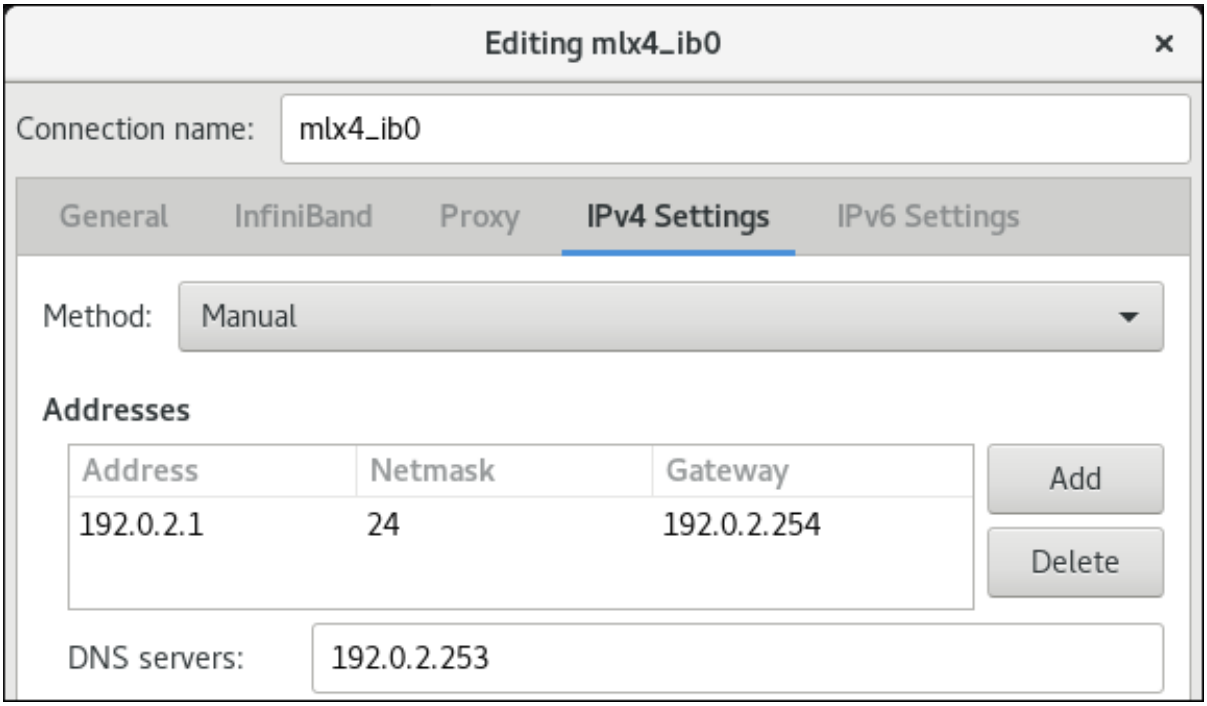
- 在服务器中安装 InfiniBand 设备，并载入相应的内核模块。
- 已安装该 **nm-connection-editor** 软件包。

流程

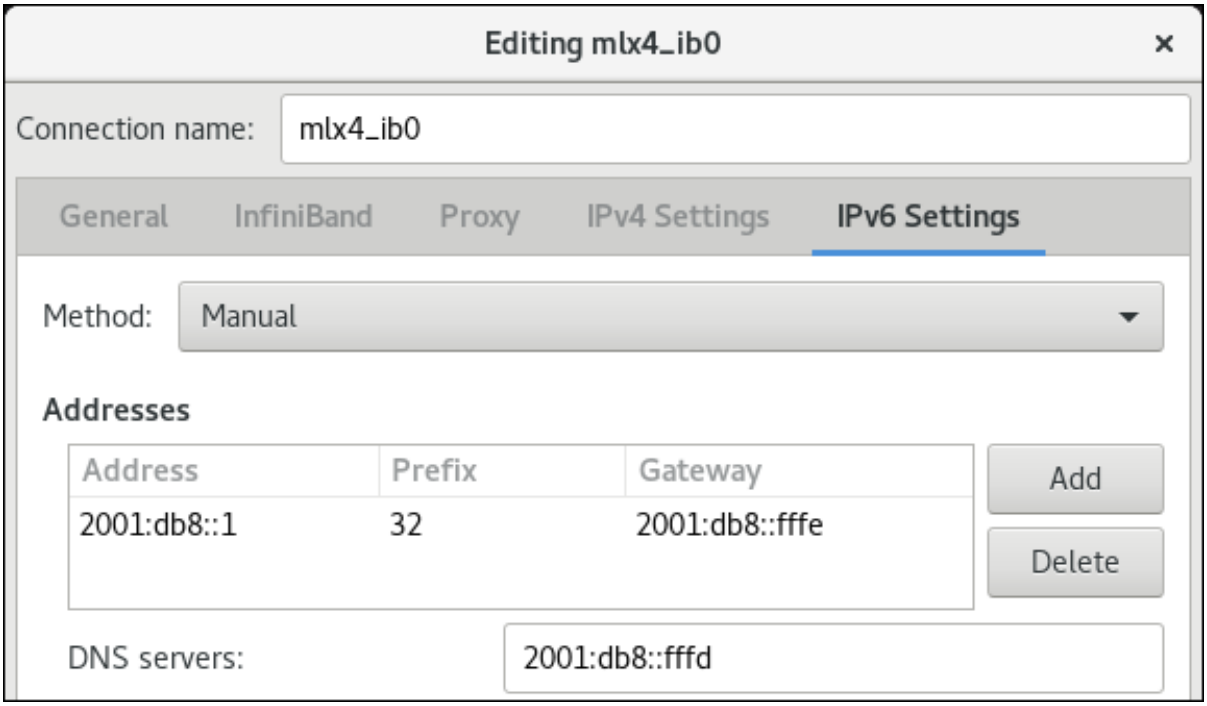
1. 打开终端窗口，输入：

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

2. 点 **+** 按钮添加新连接。
3. 选择 **InfiniBand** 连接类型并点 **Create**。
4. 在 **InfiniBand** 标签页中：
   a. （可选）更改连接名称。
   b. 选择传输模式。
   c. 选该设备。
   d. 可选：设置 MTU。
5. 在 **IPv4 Settings** 标签中配置 IPv4 设置。例如，设置静态 IPv4 地址、网络掩码、默认网关和 DNS 服务器：



6. 在 **IPv6 Settings** 标签中配置 IPv6 设置。例如，设置静态 IPv6 地址、网络掩码、默认网关和 DNS 服务器：



7. 点 **Save** 保存 team 连接。

8. 关闭 **nm-connection-editor**。

9. 可选：设置 **P_Key** 接口。请注意：您必须在命令行中设置这个参数，因为它在 **nm-connection-editor** 中不可用 。
例如，要将 **0x8002** 设置为 **mlx4_ib0** 连接的 **P_Key** 接口，请输入：

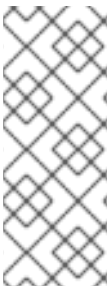
```
# nmcli connection modify mlx4_ib0 infiniband.p-key 0x8002
```

第 6 章 测试 INFINIBAND 网络

本节提供测试 InfiniBand 网络的步骤。

6.1. 测试早期 INFINIBAND RDMA 操作

这部分论述了如何测试 InfiniBand 远程直接访问(RDMA)操作。



注意

这部分只适用于 InfiniBand 设备。如果您使用基于 IP 的 iWARP 或 RoCE/IBoE 设备，请查看：

- 第 6.2 节 “使用 ping 程序测试 IPoIB”
- 第 6.3 节 “配置 IPoIB 后使用 qperf 测试 RDMA 网络”

先决条件

- 配置了 RDMA。
- 已安装 **libibverbs-utils** 和 **infiniband-diags** 软件包。

流程

1. 列出可用的 InfiniBand 设备：

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

2. 显示特定 InfiniBand 设备的信息。例如：要显示 **mlx4_1** 设备信息，,请输入：

```
# 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. 显示 InfiniBand 设备的基本状态。例如，要显示 **mlx4_1** 设备的状态，请输入：

```
# 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. 使用 **ibping** 实用程序使用 InfiniBand 从客户端 ping 到服务器：

   a. 在作为服务器的主机上，在服务器模式中启动 **ibping**：

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

   这个命令使用以下参数：

   - **-S**：启用服务器模式。
   - **-C *InfiniBand_CA_name***：设置要使用的 CA 名称。
   - **-P *port_number***：如果 InfiniBand 提供多个端口，则设置要使用的端口号。

   b. 在作为客户端的主机上，使用**ibping**，如下所示：

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

- **-c *number***：向服务器发送这些数目的数据包。
- **-C *InfiniBand_CA_name***：设置要使用的 CA 名称。
- **-P *port_number***：如果 InfiniBand 提供多个端口，则设置要使用的端口号。
- **-L *port_LID***：设置要使用的本地识别符（LID）。

其它资源

- 有关 **ibping** 参数的详情,请查看 **ibping(8)** man page。

6.2. 使用 PING 程序测试 IPOIB

配置了 IPoIB 后，使用 **ping** 工具发送 ICMP 数据包来测试 IPoIB 连接。

先决条件

- 两个 RDMA 主机在带有 RDMA 端口的同一个 InfiniBand 光纤中连接。
- 这两个主机中的 IPoIB 接口使用同一子网中的 IP 地址配置。

流程

1. 使用 **ping** 实用程序将 ICMP 数据包发送到远程主机的 InfiniBand 适配器：

   ```
   # ping -c5 192.0.2.1
   ```

   这个命令会将五个 ICMP 数据包发送到 IP 地址 **192.0.2.1**。

6.3. 配置 IPOIB 后使用 QPERF 测试 RDMA 网络

此流程描述了如何显示 InfiniBand 适配器配置,并使用 **qperf** 工具测量两个主机之间的带宽和延迟。

先决条件

- **qperf** 软件包安装在两个主机上。
- ipoib 是在两个主机上配置的。

流程

1. **qperf** 在没有选项作为服务器的主机中启动：

   ```
   # qperf
   ```

2. 在客户端中运行以下命令。这些命令使用客户端的 **mlx4_0** 主机频道适配器的端口 **1** 来连接到分配给服务器中 InfiniBand 适配器的 IP 地址 **192.0.2.1**。

   a. 要显示配置，请输入：

```
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. 要显示可靠连接(RC)流传输双向带宽,请输入：

```
# 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. 要显示 RC 流单向带宽,请输入：

```
# 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
------------------------
```

其它资源

- 有关详情请参考 **qperf qperf(1)** man page。