



Red Hat Enterprise Linux 8

Configuração das redes InfiniBand e RDMA

Um guia para configurar as redes InfiniBand e RDMA no Red Hat Enterprise Linux 8

Red Hat Enterprise Linux 8 Configuração das redes InfiniBand e RDMA

Um guia para configurar as redes InfiniBand e RDMA no Red Hat Enterprise Linux 8

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.

Nota Legal

Copyright © 2021 | You need to change the HOLDER entity in the en-US/Configuring_InfiniBand_and_RDMA_networks.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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] 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[®] 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.

Resumo

Este documento descreve o que são InfiniBand e acesso remoto direto à memória (RDMA) e como configurar o hardware InfiniBand. Além disso, esta documentação explica como configurar os serviços relacionados ao InfiniBand.

Índice

FORNECENDO FEEDBACK SOBRE A DOCUMENTAÇÃO DA RED HAT	3
CAPÍTULO 1. ENTENDENDO INFINIBAND E RDMA	4
Recursos adicionais	4
CAPÍTULO 2. CONFIGURAÇÃO DO ROCE	5
2.1. VISÃO GERAL DAS VERSÕES DO PROTOCOLO ROCE	5
Recursos adicionais	5
2.2. ALTERAÇÃO TEMPORÁRIA DA VERSÃO PADRÃO ROCE	5
Pré-requisitos	5
Procedimento	6
2.3. CONFIGURANDO O SOFT-ROCE	6
Procedimento	6
CAPÍTULO 3. CONFIGURAÇÃO DO SUBSISTEMA CENTRAL RDMA	8
3.1. CONFIGURANDO O SERVIÇO RDMA	8
Procedimento	8
3.2. RENOMEANDO OS DISPOSITIVOS IPOIB	8
Pré-requisitos	8
Procedimento	8
Recursos adicionais	9
3.3. AUMENTAR A QUANTIDADE DE MEMÓRIA QUE OS USUÁRIOS TÊM PERMISSÃO DE FIXAR NO SISTEMA	9
CAPÍTULO 4. CONFIGURAÇÃO DE UM GERENCIADOR DE SUB-REDE INFINIBAND	11
4.1. INSTALANDO O GERENCIADOR DE SUB-REDE OPENSMD	11
4.2. CONFIGURAÇÃO DO OPENSMD USANDO O MÉTODO SIMPLES	11
4.3. CONFIGURAÇÃO DO OPENSMD ATRAVÉS DA EDIÇÃO DO ARQUIVO OPENSMD.CONF	12
4.4. CONFIGURAÇÃO DE MÚLTIPLAS INSTÂNCIAS OPENSMD	13
4.5. CRIAÇÃO DE UMA CONFIGURAÇÃO DE PARTIÇÃO	14
CAPÍTULO 5. CONFIGURANDO O IPOIB	17
5.1. OS MODOS DE COMUNICAÇÃO IPOIB	17
5.2. ENTENDENDO OS ENDEREÇOS DE HARDWARE IPOIB	17
Recursos adicionais	18
5.3. CONFIGURAÇÃO DE UMA CONEXÃO IPOIB USANDO COMANDOS NMCLI	18
Pré-requisitos	18
Procedimento	18
5.4. CONFIGURAÇÃO DE UMA CONEXÃO IPOIB USANDO UM EDITOR DE NM-CONEXÃO	19
Pré-requisitos	19
Procedimento	19
CAPÍTULO 6. TESTE DE REDES INFINIBAND	21
6.1. TESTE DAS PRIMEIRAS OPERAÇÕES INFINIBAND RDMA	21
Pré-requisitos	21
Procedimento	21
Recursos adicionais	23
6.2. TESTE DE UM IPOIB USANDO O UTILITÁRIO PING	23
6.3. TESTE DE UMA REDE RDMA USANDO QPERF APÓS IPOIB SER CONFIGURADO	23

FORNECENDO FEEDBACK SOBRE A DOCUMENTAÇÃO DA RED HAT

Agradecemos sua contribuição em nossa documentação. Por favor, diga-nos como podemos melhorá-la. Para fazer isso:

- Para comentários simples sobre passagens específicas:
 1. Certifique-se de que você está visualizando a documentação no formato *Multi-page HTML*. Além disso, certifique-se de ver o botão **Feedback** no canto superior direito do documento.
 2. Use o cursor do mouse para destacar a parte do texto que você deseja comentar.
 3. Clique no pop-up **Add Feedback** que aparece abaixo do texto destacado.
 4. Siga as instruções apresentadas.
- Para enviar comentários mais complexos, crie um bilhete Bugzilla:
 1. Ir para o site da [Bugzilla](#).
 2. Como Componente, use **Documentation**.
 3. Preencha o campo **Description** com sua sugestão de melhoria. Inclua um link para a(s) parte(s) relevante(s) da documentação.
 4. Clique em **Submit Bug**.

CAPÍTULO 1. ENTENDENDO INFINIBAND E RDMA

InfiniBand se refere a duas coisas distintas:

- O protocolo de camada de ligação física para redes InfiniBand
- O InfiniBand Verbs API, que é uma implementação da tecnologia de acesso remoto direto à memória (RDMA)

RDMA fornece acesso à memória de um computador para a memória de outro computador sem envolver o sistema operacional de nenhum dos computadores. Esta tecnologia permite uma rede de alto rendimento e baixa latência com baixa utilização da CPU.

Em uma típica transferência de dados IP, quando uma aplicação em uma máquina envia dados para uma aplicação em outra máquina, o seguinte acontece no lado do receptor:

1. O núcleo deve receber os dados.
2. O núcleo deve determinar que os dados pertencem à aplicação.
3. O núcleo acorda a aplicação.
4. O núcleo espera que a aplicação execute uma chamada de sistema para o núcleo.
5. A aplicação copia os dados do próprio espaço de memória interna do kernel para o buffer fornecido pela aplicação.

Este processo significa que a maioria do tráfego de rede é copiada através da memória principal do sistema se o adaptador host usar acesso direto à memória (DMA), ou pelo menos duas vezes. Além disso, o computador executa uma série de trocas de contexto para alternar entre o kernel e o contexto da aplicação. Ambas as trocas de contexto podem causar uma alta carga de CPU com altas taxas de tráfego e diminuir a velocidade de outras tarefas.

A comunicação RDMA ultrapassa a intervenção do kernel no processo de comunicação, ao contrário da comunicação IP normal. Isto reduz a sobrecarga da CPU. O protocolo RDMA permite que o adaptador host saiba quando um pacote chega da rede, qual aplicação deve recebê-lo e onde, no espaço de memória da aplicação, o pacote deve ser armazenado. Em vez de enviar o pacote para o kernel para ser processado e depois copiado na memória da aplicação do usuário, com InfiniBand, o adaptador host coloca o conteúdo do pacote diretamente no buffer da aplicação. Este processo requer uma API separada, a InfiniBand Verbs API, e as aplicações devem suportar esta API antes de poderem usar o RDMA.

O Red Hat Enterprise Linux 8 suporta tanto o hardware InfiniBand quanto o InfiniBand Verbs API. Além disso, o Red Hat Enterprise Linux suporta as seguintes tecnologias que permitem o uso da API InfiniBand Verbs em hardware não InfiniBand:

- Internet Wide Area RDMA Protocol (iWARP): Um protocolo de rede que implementa o RDMA sobre redes IP.
- RDMA sobre Ethernet convergente (RoCE), também conhecida como InfiniBand over Ethernet (IBoE): Um protocolo de rede que implementa as redes RDMA sobre Ethernet.

Recursos adicionais

- Para detalhes sobre a implementação de um software RoCE, veja [Capítulo 2, Configuração do RoCE](#).

CAPÍTULO 2. CONFIGURAÇÃO DO ROCE

Esta seção explica informações de fundo sobre RDMA sobre Ethernet convergente (RoCE), bem como como alterar a versão padrão RoCE, e como configurar um adaptador de software RoCE.

Observe que existem diferentes fornecedores, tais como Mellanox, Broadcom e QLogic, que fornecem hardware RoCE.

2.1. VISÃO GERAL DAS VERSÕES DO PROTOCOLO ROCE

O RoCE é um protocolo de rede que permite o acesso remoto direto à memória (RDMA) via Ethernet.

A seguir são apresentadas as diferentes versões RoCE:

RoCE v1

O protocolo RoCE versão 1 é um protocolo de camada de link Ethernet com ethertype **0x8915** que permite a comunicação entre quaisquer dois hosts no mesmo domínio de transmissão Ethernet.

Por default, ao usar um adaptador de rede Mellanox ConnectX-3, o Red Hat Enterprise Linux usa o RoCE v1 para o RDMA Connection Manager (RDMA_CM).

RoCE v2

O protocolo RoCE versão 2 existe sobre o protocolo UDP sobre IPv4 ou o UDP sobre IPv6. A porta de destino UDP número 4791 é reservada para o RoCE v2.

Por default, ao usar um adaptador de rede Mellanox ConnectX-3 Pro, ConnectX-4 Lx ou ConnectX-5, o Red Hat Enterprise Linux usa o RoCE v2 para o RDMA_CM, mas o hardware suporta tanto o RoCE v1 quanto o RoCE v2.

O RDMA_CM estabelece uma conexão confiável entre um cliente e um servidor para a transferência de dados. O RDMA_CM fornece uma interface RDMA neutra em termos de transporte para estabelecer conexões. A comunicação usa um dispositivo RDMA específico, e as transferências de dados são baseadas em mensagens.



IMPORTANTE

O uso de RoCE v2 no cliente e RoCE v1 no servidor não é suportado. Neste caso, configure tanto o servidor quanto o cliente para se comunicar através da RoCE v1.

Recursos adicionais

- [Seção 2.2, "Alteração temporária da versão padrão RoCE"](#)

2.2. ALTERAÇÃO TEMPORÁRIA DA VERSÃO PADRÃO ROCE

O uso do protocolo RoCE v2 no cliente e RoCE v1 no servidor não é suportado. Se o hardware em seu servidor suporta apenas o RoCE v1, configure seus clientes para se comunicar com o servidor usando o RoCE v1. Esta seção descreve como aplicar o RoCE v1 no cliente que usa o driver **mlx5_0** para o dispositivo Mellanox ConnectX-5 Infiniband. Observe que as mudanças descritas nesta seção são apenas temporárias até que você reinicialize o host.

Pré-requisitos

- O cliente utiliza um dispositivo InfiniBand que utiliza, por padrão, o protocolo RoCE v2.

- O dispositivo InfiniBand no servidor suporta apenas o RoCE v1.

Procedimento

1. Criar o `/sys/kernel/config/rdma_cm/mlx5_0/` diretório:

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

2. Exibir o modo RoCE padrão. Por exemplo, para exibir o modo para a porta 1:

```
# cat /sys/kernel/config/rdma_cm/mlx5_0/ports/1/default_roce_mode
RoCE v2
```

3. Mude o modo RoCE padrão para a versão 1:

```
# echo {\i1}"IB/RoCE v1}" > /sys/kernel/config/rdma_cm/mlx5_0/ports/1/default_roce_mode
```

2.3. CONFIGURANDO O SOFT-ROCE

Soft-RoCE é uma implementação de software de acesso remoto direto à memória (RDMA) sobre Ethernet, que também é chamado de RXE. Esta seção descreve como configurar o Soft-RoCE.

Use Soft-RoCE em hosts sem adaptadores de canal host RoCE (HCA).

Pré-requisitos

- Um adaptador Ethernet é instalado no sistema.

Procedimento

1. Instale os pacotes **libibverbs**, **libibverbs-utils**, e **infiniband-diags**:

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

2. Carregue o módulo do kernel **rdma_rxe** e exiba a configuração atual:

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

3. Acrescentar um novo dispositivo RXE. Por exemplo, para adicionar o dispositivo Ethernet **enp7s0** como um dispositivo RXE, entre:

```
# rxe_cfg adicionar enp7s0
```

4. Exibir o status do dispositivo RXE:

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

Na coluna **RDEV**, você vê que o **enp7s0** está mapeado para o dispositivo **rxe0**.

5. Opcional: liste os dispositivos RDMA disponíveis no sistema:

```
# ibv_devices
device      node GUID
-----
rxe0       505400ffed5e0fb
```

Alternativamente, use o utilitário **ibstat** para exibir um status detalhado:

```
# ibstat rxe0
CA 'rxe0'
CA type:
Number of ports: 1
Firmware version:
Hardware version:
Node GUID: 0x505400ffed5e0fb
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: 0x505400ffed5e0fb
Link layer: Ethernet
```

CAPÍTULO 3. CONFIGURAÇÃO DO SUBSISTEMA CENTRAL RDMA

Esta seção descreve como configurar o serviço **rdma** e aumentar a quantidade de memória que os usuários têm permissão de fixar no sistema.

3.1. CONFIGURANDO O SERVIÇO RDMA

O serviço **rdma** gerencia a pilha RDMA no núcleo. Se o Red Hat Enterprise Linux detectar dispositivos InfiniBand, iWARP, ou RoCE, o gerente do dispositivo **udev** instrui **systemd** a iniciar o serviço **rdma**.

Procedimento

1. Edite o arquivo **/etc/rdma/rdma.conf** e defina as variáveis dos módulos que você deseja habilitar para **yes**. O seguinte é o padrão **/etc/rdma/rdma.conf** no Red Hat Enterprise Linux 8:

```
# 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. Reinicie o serviço **rdma**:

```
# systemctl restart rdma
```

3.2. RENOMEANDO OS DISPOSITIVOS IPOIB

Por padrão, os nomes dos kernel IP sobre dispositivos InfiniBand (IPoIB), por exemplo, **ib0**, **ib1**, e assim por diante. Para evitar conflitos, a Red Hat recomenda a criação de uma regra no gerenciador de dispositivos **udev** para criar nomes persistentes e significativos, tais como **mlx4_ib0**.

Pré-requisitos

- Um dispositivo InfiniBand é instalado no host.

Procedimento

1. Mostrar o endereço do hardware do dispositivo. Por exemplo, para exibir o endereço do dispositivo chamado **ib0**, digite:

2. Use o comando **ulimit -l** para exibir o limite:

```
┆ $ ulimit -l  
┆ unlimited
```

Se o comando retornar **unlimited**, o usuário pode fixar uma quantidade ilimitada de memória.

Recursos adicionais

- Para mais detalhes sobre a limitação de recursos do sistema, consulte a página de manual **limits.conf(5)**.

CAPÍTULO 4. CONFIGURAÇÃO DE UM GERENCIADOR DE SUB-REDE INFINIBAND

Todas as redes InfiniBand devem ter um gerenciador de sub-rede em funcionamento para que a rede funcione. Isto é verdade mesmo que duas máquinas estejam conectadas diretamente, sem nenhum switch envolvido.

É possível ter mais de um gerente de sub-rede. Nesse caso, um atua como um master e outro atua como um escravo que assumirá o controle caso o master falhe.

A maioria dos interruptores InfiniBand contém um gerenciador de sub-rede incorporado. Entretanto, se você precisar de um gerenciador de sub-rede mais atualizado ou se precisar de mais controle, use o gerenciador de sub-rede **OpenSM** fornecido pelo Red Hat Enterprise Linux.

4.1. INSTALANDO O GERENCIADOR DE SUB-REDE OPENSM

Esta seção descreve como instalar o gerenciador de sub-rede OpenSM.

Procedimento

1. Instale o pacote **opensm**:

```
# yum instalar opensm
```

2. Configure o OpenSM se a instalação padrão não for compatível com seu ambiente. Se apenas uma porta InfiniBand estiver instalada, o host deverá atuar como o gerente da subrede mestre, e nenhuma mudança personalizada é necessária. A configuração padrão funciona sem nenhuma modificação.

3. Habilite e inicie o serviço **opensm**:

```
# systemctl habilitado --agora aberto
```

Recursos adicionais

- Para uma lista de opções de linha de comando para o serviço **opensm**, bem como descrições adicionais das configurações das partições, Qualidade de Serviço (QoS) e outros tópicos avançados, consulte a página de manual **opensm(8)**.

4.2. CONFIGURAÇÃO DO OPENSM USANDO O MÉTODO SIMPLES

Esta seção descreve como configurar o OpenSM se você não precisar de nenhuma configuração personalizada.

Pré-requisitos

- Uma ou mais portas InfiniBand são instaladas no servidor.

Procedimento

1. Obter os GUIDs para os portos usando o utilitário **ibstat**:

```
# 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
```



NOTA

Alguns adaptadores InfiniBand usam o mesmo GUID para o nó, sistema e porta.

2. Edite o arquivo `/etc/sysconfig/opensm` e defina os GUIDs no parâmetro **GUIDS**:

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

3. Opcionalmente, defina o parâmetro **PRIORITY** se vários gerentes de sub-rede estiverem disponíveis em sua sub-rede. Por exemplo:

```
PRIORIDADE=15
```

Recursos adicionais

- Para informações adicionais sobre os parâmetros que você pode definir em `/etc/sysconfig/opensm`, veja a documentação nesse arquivo.

4.3. CONFIGURAÇÃO DO OPENSMM ATRAVÉS DA EDIÇÃO DO ARQUIVO OPENSMM.CONF

Esta seção descreve como configurar o OpenSM, editando o arquivo `/etc/rdma/opensm.conf`. Use este método para personalizar a configuração do OpenSM se apenas uma porta InfiniBand estiver disponível.

Pré-requisitos

- Apenas uma porta InfiniBand está instalada no servidor.

Procedimento

1. Edite o arquivo `/etc/rdma/opensm.conf` e personalize as configurações de acordo com seu ambiente.
2. Reinicie o serviço **opensm**:

```
# systemctl restart opensm
```

Recursos adicionais

- Quando você instala um pacote **opensm** atualizado, o utilitário **yum** armazena o novo arquivo de configuração do OpenSM como `/etc/rdma/opensm.conf.rpmnew`. Compare este arquivo com seu arquivo personalizado `/etc/rdma/opensm.conf` e incorpore manualmente as mudanças.

4.4. CONFIGURAÇÃO DE MÚLTIPLAS INSTÂNCIAS OPENSMS

Esta seção descreve como configurar múltiplas instâncias de OpenSM.

Pré-requisitos

- Uma ou mais portas InfiniBand são instaladas no servidor.

Procedimento

1. Opcionalmente, copiar o arquivo `/etc/rdma/opensm.conf` para o arquivo `/etc/rdma/opensm.conf.orig`:

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

Quando você instala um pacote **opensm** atualizado, o utilitário **yum** substitui o `/etc/rdma/opensm.conf`. Com a cópia criada nesta etapa, você pode comparar o arquivo anterior com o novo para identificar as alterações e incorporá-las manualmente nos arquivos **opensm.conf** específicos da instância.

2. Crie uma cópia do arquivo `/etc/rdma/opensm.conf`:

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

Para cada instância criada, anexe um número único e contínuo à cópia do arquivo de configuração.

3. Edite a cópia que você criou na etapa anterior e personalize as configurações da instância para adequá-la ao seu ambiente. Por exemplo, defina os parâmetros **guid**, **subnet_prefix**, e **logdir**.
4. Opcionalmente, criar um arquivo **partitions.conf** com um nome exclusivo especificamente para esta sub-rede e referenciar esse arquivo no parâmetro **partition_config_file**, na cópia correspondente do arquivo **opensm.conf**.

5. Repita os passos anteriores para cada instância que você deseja criar.
6. Iniciar o serviço **opensm**:

```
# systemctl start opensm
```

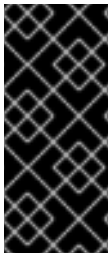
O serviço **opensm** inicia automaticamente uma instância única para cada arquivo **opensm.conf.*** no diretório **/etc/rdma/**. Caso existam vários arquivos **opensm.conf.***, o serviço ignora as configurações no arquivo **/etc/sysconfig/opensm**, bem como no arquivo base **/etc/rdma/opensm.conf**.

Recursos adicionais

- Quando você instala um pacote **opensm** atualizado, o utilitário **yum** armazena o novo arquivo de configuração do OpenSM como **/etc/rdma/opensm.conf.rpmnew**. Compare este arquivo com seus arquivos personalizados **/etc/rdma/opensm.conf.*** e incorpore manualmente as mudanças.

4.5. CRIAÇÃO DE UMA CONFIGURAÇÃO DE PARTIÇÃO

Esta seção descreve como criar configurações de partição InfiniBand para OpenSM. As partições permitem aos administradores criar sub-redes em InfiniBand similares às VLANs Ethernet.



IMPORTANTE

Se você definir uma partição com uma velocidade específica, como 40 Gbps, todos os hosts dentro desta partição devem suportar pelo menos esta velocidade. Se um host não atender aos requisitos de velocidade, ele não poderá aderir à partição. Portanto, defina a velocidade de uma partição para a velocidade mais baixa suportada por qualquer host com permissão para entrar na partição.

Pré-requisitos

- Uma ou mais portas InfiniBand são instaladas no servidor.

Procedimento

1. Edite o arquivo **/etc/rdma/partitions.conf** e configure as partições.



NOTA

Todos os tecidos devem conter a divisória **0x7fff**, e todos os interruptores e todos os hospedeiros devem pertencer a esse tecido.

Por exemplo, adicione o seguinte conteúdo ao arquivo para criar a partição padrão **0x7fff** a uma velocidade reduzida de 10 Gbps, e uma partição **0x0002** com uma velocidade de 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;
```

CAPÍTULO 5. CONFIGURANDO O IPOIB

Por padrão, o InfiniBand não utiliza o protocolo de Internet (IP) para comunicação. Entretanto, o IP sobre InfiniBand (IPoIB) fornece uma camada de emulação de rede IP sobre as redes de acesso remoto direto à memória (RDMA) InfiniBand. Isto permite que aplicações existentes não modificadas transmitam dados sobre redes InfiniBand, mas o desempenho é menor do que se a aplicação usasse RDMA nativamente.



NOTA

As redes Internet Wide Area RDMA Protocol (iWARP) e RoCE já são baseadas em IP. Portanto, não se pode criar um dispositivo IPoIB em cima de dispositivos iWARP ou RoCE.

5.1. OS MODOS DE COMUNICAÇÃO IPOIB

Você pode configurar um dispositivo IPoIB no modo **Datagram** ou **Connected**. A diferença é o tipo de par de filas que a camada IPoIB tenta abrir com a máquina na outra extremidade da comunicação:

- No modo **Datagram**, o sistema abre um par de filas desconectado e não confiável. Este modo não suporta pacotes maiores que a Unidade Máxima de Transmissão (MTU) da camada de ligação InfiniBand. A camada IPoIB adiciona um cabeçalho IPoIB de 4 bytes no topo do pacote IP que está sendo transmitido. Como resultado, a MTU IPoIB deve ser 4 bytes menor do que a MTU da camada de link InfiniBand. Como 2048 é um MTU de camada de link InfiniBand comum, o MTU do dispositivo IPoIB comum no modo **Datagram** é 2044.
- No modo **Connected**, o sistema abre um par de filas conectado e confiável. Este modo permite mensagens maiores que a MTU de camada de link InfiniBand, e o adaptador do host trata da segmentação e remontagem de pacotes. Como resultado, não há limite de tamanho imposto ao tamanho das mensagens IPoIB que podem ser enviadas pelos adaptadores InfiniBand no modo **Connected**. Entretanto, os pacotes IP são limitados por causa do campo **size** e dos cabeçalhos TCP/IP. Por este motivo, o MTU IPoIB no modo **Connected** é de no máximo **65520** bytes.

O modo **Connected** tem um desempenho superior, mas consome mais memória do kernel.

Se um sistema é configurado para usar o modo **Connected**, ele ainda envia tráfego multicast no modo **Datagram**, porque as chaves InfiniBand e o tecido não podem passar tráfego multicast no modo **Connected**. Além disso, o sistema volta ao modo **Datagram**, quando se comunica com qualquer host que não esteja configurado no modo **Connected**.

Ao executar a aplicação que envia dados multicast até o MTU máximo na interface, você deve configurar a interface no modo **Datagram** ou configurar a aplicação para limitar o tamanho do envio de pacotes a um tamanho que caberá em pacotes do tamanho de datagramas.

5.2. ENTENDENDO OS ENDEREÇOS DE HARDWARE IPOIB

Os dispositivos IPoIB têm um endereço de hardware de 20 bytes que consiste das seguintes partes:

- Os primeiros 4 bytes são bandeiras e números de pares de filas.
- Os próximos 8 bytes são o prefixo da sub-rede. O prefixo padrão da sub-rede é **0xfe:80:00:00:00:00:00**. Após o dispositivo se conectar ao gerenciador de sub-rede, o dispositivo muda este prefixo para corresponder ao configurado no gerenciador de sub-rede.

- Os últimos 8 bytes são o Identificador Global Único (GUID) da porta InfiniBand a que o dispositivo IPoIB está anexado.



NOTA

Como os primeiros 12 bytes podem mudar, não os utilize nas regras do gerenciador de dispositivos **udev**.

Recursos adicionais

- Para detalhes sobre como renomear os dispositivos IPoIB, veja [Seção 3.2, “Renomeando os dispositivos IPoIB”](#).

5.3. CONFIGURAÇÃO DE UMA CONEXÃO IPOIB USANDO COMANDOS NMCLI

Este procedimento descreve como configurar uma conexão IPoIB usando os comandos **nmcli**.

Pré-requisitos

- Um dispositivo InfiniBand é instalado no servidor, e o módulo do kernel correspondente é carregado.

Procedimento

1. Criar a conexão InfiniBand. Por exemplo, para criar uma conexão que utilize a interface **mlx4_ib0** no modo de transporte **Connected** e o máximo de MTU de **65520** bytes, entre:

```
# nmcli conexão adicionar tipo infiniband con-name mlx4_ib0 ifname mlx4_ib0 modo de
transporte Connected mtu 65520
```

2. Opcional: definir uma interface **P_Key**. Por exemplo, para definir **0x8002** como **P_Key** interface da conexão **mlx4_ib0**, entre:

```
# nmcli conexão modificar mlx4_ib0 infiniband.p-key 0x8002
```

3. Configurar as configurações do IPv4. Por exemplo, para configurar um endereço IPv4 estático, máscara de rede, gateway padrão e servidor DNS da conexão **mlx4_ib0**, entre:

```
# 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. Configurar as configurações IPv6. Por exemplo, para configurar um endereço IPv6 estático, máscara de rede, gateway padrão e servidor DNS da conexão **mlx4_ib0**, entre:

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

5. Ativar a conexão. Por exemplo, para ativar a conexão **mlx4_ib0**:

```
# nmcli conexão up mlx4_ib0
```

5.4. CONFIGURAÇÃO DE UMA CONEXÃO IPOIB USANDO UM EDITOR DE NM-CONEXÃO

Este procedimento descreve como configurar uma conexão IPOIB usando a aplicação **nm-connection-editor**.


Pré-requisitos

- Um dispositivo InfiniBand é instalado no servidor, e o módulo do kernel correspondente é carregado.
- O pacote **nm-connection-editor** está instalado.

Procedimento

1. Abra um terminal, e entre:

```
Monitor de conexão de $ nm
```

2. Clique no botão  para adicionar uma nova conexão.
3. Selecione o tipo de conexão **InfiniBand**, e clique em **Criar**.
4. Na aba **InfiniBand**:
 - a. Opcionalmente, mudar o nome da conexão.
 - b. Selecione o modo de transporte.
 - c. Selecione o dispositivo.
 - d. Opcional: definir uma MTU.

- Na aba **IPv4 Settings**, configure as configurações do IPv4. Por exemplo, defina um endereço IPv4 estático, máscara de rede, gateway padrão e servidor DNS

Editing mlx4_ib0

Connection name:

General InfiniBand Proxy **IPv4 Settings** IPv6 Settings

Method:

Addresses

Address	Netmask	Gateway
192.0.2.1	24	192.0.2.254

DNS servers:

- Na aba **IPv6 Settings**, configure as configurações IPv6. Por exemplo, configure um endereço IPv6 estático, máscara de rede, gateway padrão e servidor DNS

Editing mlx4_ib0

Connection name:

General InfiniBand Proxy IPv4 Settings **IPv6 Settings**

Method:

Addresses

Address	Prefix	Gateway
2001:db8::1	32	2001:db8::fffe

DNS servers:

- Clique em **Salvar** para salvar a conexão da equipe.
- Fechar **nm-connection-editor**.
- Opcional: definir uma interface **P_Key**. Note que você deve definir este parâmetro na linha de comando, pois a configuração não está disponível em **nm-connection-editor**.
Por exemplo, para definir **0x8002** como **P_Key** interface da conexão **mlx4_ib0**, entre:

```
# nmcli conexão modificar mlx4_ib0 infiniband.p-key 0x8002
```


CAPÍTULO 6. TESTE DE REDES INFINIBAND

Esta seção fornece procedimentos para testar as redes InfiniBand.

6.1. TESTE DAS PRIMEIRAS OPERAÇÕES INFINIBAND RDMA

Esta seção descreve como testar as operações de acesso remoto direto à memória InfiniBand (RDMA).



NOTA

Esta seção se aplica apenas aos dispositivos InfiniBand. Se você usa dispositivos iWARP ou RoCE/IBoE, que são baseados em IP, veja:

- [Seção 6.2, “Teste de um IPoIB usando o utilitário ping”](#)
- [Seção 6.3, “Teste de uma rede RDMA usando qperf após IPoIB ser configurado”](#)

Pré-requisitos

- O RDMA está configurado.
- Os pacotes **libibverbs-utils** e **infiniband-diags** estão instalados.

Procedimento

1. Liste os dispositivos InfiniBand disponíveis:

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

2. Exibir as informações para um dispositivo InfiniBand específico. Por exemplo, para exibir as informações do dispositivo **mlx4_1**, entre:

```
# 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. Exibir o status básico de um dispositivo InfiniBand. Por exemplo, para exibir o status do dispositivo **mlx4_1**, entre:

```

# 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: 0xf65214ffe7bcba2
  Link layer: Ethernet

```

4. Use o utilitário **ibping** para pingar de um cliente para um servidor usando InfiniBand:
- No host que atua como um servidor, inicie **ibping** no modo servidor:

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

Este comando usa os seguintes parâmetros:

- **-S**: Habilita o modo servidor.
- **-C *InfiniBand_CA_name***: Definir é o nome CA a ser usado.

- **-P *port_number***. Define o número da porta a ser utilizada, se a InfiniBand fornecer várias portas.
- b. No anfitrião que atua como cliente, use **ibping** como segue:

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

- **-c *number***. Envia este número de pacotes para o servidor.
- **-C *InfiniBand_CA_name***: Definir é o nome CA a ser usado.
- **-P *port_number***. Define o número da porta a ser utilizada, se a InfiniBand fornecer várias portas.
- **-L *port_LID***. Define o identificador local (LID) a ser utilizado.

Recursos adicionais

- Para mais detalhes sobre os parâmetros **ibping**, consulte a página de manual **ibping(8)**.

6.2. TESTE DE UM IPOIB USANDO O UTILITÁRIO PING

Depois de configurar o IPoIB, use o utilitário **ping** para enviar pacotes ICMP para testar a conexão IPoIB.

Pré-requisitos

- Os dois anfitriões RDMA estão conectados no mesmo tecido InfiniBand com portas RDMA.
- As interfaces IPoIB em ambos os hosts são configuradas com endereços IP dentro da mesma sub-rede.

Procedimento

1. Use o utilitário **ping** para enviar pacotes ICMP para o adaptador InfiniBand do host remoto:

```
# ping -c5 192.0.2.1
```

Este comando envia cinco pacotes ICMP para o endereço IP **192.0.2.1**.

6.3. TESTE DE UMA REDE RDMA USANDO QPERF APÓS IPOIB SER CONFIGURADO

Este procedimento descreve exemplos de como exibir a configuração do adaptador InfiniBand e medir a largura de banda e a latência entre dois hosts usando o utilitário **qperf**.

Pré-requisitos

- O pacote **qperf** está instalado em ambos os hosts.
- O IPoIB é configurado em ambos os hosts.

Procedimento

1. Inicie **qperf** em um dos hosts sem nenhuma opção para atuar como servidor:

```
# qperf
```

2. Use os seguintes comandos sobre o cliente. Os comandos utilizam a porta **1** do adaptador do canal host **mlx4_0** no cliente para conectar ao endereço IP **192.0.2.1** atribuído ao adaptador InfiniBand no servidor.

- a. Para exibir a configuração, entre:

```
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. Para exibir a Conexão Confiável (RC) com largura de banda bidirecional, entre:

```
# 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. Para exibir o RC streaming de largura de banda unidirecional, entre:

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

Recursos adicionais

- Para mais detalhes sobre **qperf**, consulte a página de manual **qperf(1)**.

