最も一般的な手法: ファイアウォールによってノード間の MPI 通信がブロックされた場合の対応方法

この記事は、インテル® デベロッパー・ゾーンに公開されている「Best Known Methods: Firewall Blocks MPI Communication among Nodes」の日本語参考訳です。

この記事では、ファイアウォールによって複数のマシン間のメッセージ・パッシング・インターフェイス (MPI) 通信がブロックされた場合に有効な 3 つの手法を紹介します。例えば、2 つのマシン間で MPI プログラムを実行する場合、次のような通信エラーが発生することがあります。

```
[proxy:0:1@knl-sb0] HYDU_sock_connect (../../utils/sock/sock.c:268): unable to
connect from "knl-sb0" to "knc4" (No route to host)
[proxy:0:1@knl-sb0] main (../../pm/pmiserv/pmip.c:461): unable to connect to
server knc4 at port 39652 (check for firewalls!)
```

これは、ファイアウォールによって MPI 通信がブロックされ、MPI ランクが互いに通信できないことを示しています。

以下の 3 つの手法は、この問題に対応するのに役立ちます。

手法 1: `firewalld` デーモンを停止する

最も単純な手法は、MPI プログラムを実行するマシンでファイアウォールを停止することです。ここでは、最初に Red Hat* Enterprise Linux* (RHEL) と CentOS* システムで `firewalld` デーモンのステータスをチェックします。

```
$ systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor
preset: enabled)
   Active: active (running) since Tue 2017-12-05 21:36:10 PST; 12min ago
 Main PID: 47030 (firewalld)
   CGroup: /system.slice/firewalld.service
           47030 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid
```

出力から `firewalld` が動作していることが分かります。次のコマンドを実行して、デーモンを停止し、ステータスを確認できます。

```
$ sudo systemctl stop firewalld
$ systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor
preset: enabled)
   Active: inactive (dead) since Tue 2017-12-05 21:51:19 PST; 4s ago
  Process: 48062 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS
(code=exited, status=0/SUCCESS)
 Main PID: 48062 (code=exited, status=0/SUCCESS)
```

`firewalld` を停止したことで、2 つのマシン間で MPI プログラムを実行できるようになります (この例では、MPI プログラムとしてインテル® MPI Benchmarks (英語) の `IMB-MPI1` を使用)。

```
$ mpirun -host localhost -n 1 /opt/intel/impi/2018.0.128/bin64/IMB-MPI1
Sendrecv : -host 10.23.3.61 -n 1  /opt/intel/impi/2018.0.128/bin64/IMB-MPI1
#------------------------------------------------------------
#    Intel (R) MPI Benchmarks 2018, MPI-1 part
#------------------------------------------------------------
# Date                  : Tue Dec  5 21:51:45 2017
# Machine               : x86_64
# System                : Linux
# Release               : 3.10.0-327.el7.x86_64
# Version               : #1 SMP Thu Nov 19 22:10:57 UTC 2015
# MPI Version           : 3.1
# MPI Thread Environment:

# Calling sequence was:

# /opt/intel/impi/2018.0.128/bin64/IMB-MPI1 Sendrecv

# Minimum message length in bytes:   0
# Maximum message length in bytes:   4194304
#
# MPI_Datatype                   :   MPI_BYTE
# MPI_Datatype for reductions    :   MPI_FLOAT
# MPI_Op                         :   MPI_SUM
#
#

# List of Benchmarks to run:

# Sendrecv

#-----------------------------------------------------------------------------
# Benchmarking Sendrecv
# #processes = 2
#-----------------------------------------------------------------------------
    #bytes #repetitions  t_min[usec]  t_max[usec]  t_avg[usec]   Mbytes/sec

         0         1000        16.57        16.57        16.57          0.00
         1         1000        16.57        16.57        16.57          0.12
         2         1000        16.52        16.53        16.53          0.24
         4         1000        16.58        16.58        16.58          0.48
         8         1000        16.51        16.51        16.51          0.97
        16         1000        16.20        16.20        16.20          1.98
        32         1000        16.32        16.32        16.32          3.92
        64         1000        16.55        16.55        16.55          7.73
       128         1000        16.65        16.65        16.65         15.37
       256         1000        29.07        29.09        29.08         17.60
       512         1000        30.75        30.76        30.76         33.29
      1024         1000        31.13        31.15        31.14         65.75
      2048         1000        33.58        33.58        33.58        121.98
      4096         1000        34.79        34.80        34.80        235.38
```

しかし、この方法では、マシンがセキュリティーの問題に対して無防備になります。シナリオによっては、適切ではないでしょう。その場合、`firewalld` デーモンを再度開始して 2 つ目の手法を試します。

```
$ sudo systemctl start firewalld
```

手法 2: `firewalld` でリッチルールを使用する

次に `firewalld` リッチルール機能を使用して、IP アドレスが 10.23.3.61 のマシンから IP v4 パケットのみを受け付けるようにします。

```
$ sudo firewall-cmd --add-rich-rule='rule family="ipv4" source
address="10.23.3.61" accept'
Success
```

追加したルールを確認します。

```
$ firewall-cmd --list-rich-rules
rule family="ipv4" source address="10.23.3.61" accept
```

MPI プログラムを実行します。

```
$ mpirun -host localhost -n 1 /opt/intel/impi/2018.0.128/bin64/IMB-MPI1
Sendrecv : -host 10.23.3.61 -n 1  /opt/intel/impi/2018.0.128/bin64/IMB-MPI1
#------------------------------------------------------------
#    Intel (R) MPI Benchmarks 2018, MPI-1 part
#------------------------------------------------------------
# Date                  : Tue Dec  5 22:01:17 2017
# Machine               : x86_64
# System                : Linux
# Release               : 3.10.0-327.el7.x86_64
# Version               : #1 SMP Thu Nov 19 22:10:57 UTC 2015
# MPI Version           : 3.1
# MPI Thread Environment:


# Calling sequence was:

# /opt/intel/impi/2018.0.128/bin64/IMB-MPI1 Sendrecv

# Minimum message length in bytes:   0
# Maximum message length in bytes:   4194304
#
# MPI_Datatype                   :   MPI_BYTE
# MPI_Datatype for reductions    :   MPI_FLOAT
# MPI_Op                         :   MPI_SUM
#
#

# List of Benchmarks to run:

# Sendrecv

#-----------------------------------------------------------------------------
# Benchmarking Sendrecv
# #processes = 2
#-----------------------------------------------------------------------------
    #bytes #repetitions  t_min[usec]  t_max[usec]  t_avg[usec]   Mbytes/sec
         0         1000        16.88        16.88        16.88         0.00
         1         1000        16.86        16.86        16.86         0.12
         2         1000        16.57        16.57        16.57         0.24
         4         1000        16.55        16.55        16.55         0.48
         8         1000        16.40        16.40        16.40         0.98
        16         1000        16.29        16.29        16.29         1.96
        32         1000        16.63        16.63        16.63         3.85
        64         1000        16.87        16.87        16.87         7.59
       128         1000        17.03        17.04        17.03        15.03
       256         1000        27.58        27.60        27.59        18.55
       512         1000        27.52        27.54        27.53        37.18
      1024         1000        26.87        26.89        26.88        76.16
      2048         1000        28.62        28.64        28.63       143.02
      4096         1000        30.27        30.27        30.27       270.62

^C[mpiexec@knc4] Sending Ctrl-C to processes as requested
[mpiexec@knc4] Press Ctrl-C again to force abort
```

次のコマンドを実行して、定義したリッチルールを削除できます。

```
$ sudo firewall-cmd --remove-rich-rule='rule family="ipv4" source
address="10.23.3.61" accept'
success
```

手法 3: iptables-service にルールを追加して特定のマシンからパケットを受け付ける

firewalld に加えて、iptables-service も RHEL と CentOS* システムでファイアウォールを管理するのに使用されます。この手法では、iptables-service にルールを追加して、特定のマシンからのトラフィックのみを許可します。

最初に、iptables-services パッケージをダウンロードしてインストールします。

```
$ sudo yum install iptables-servicesi
```

次に、iptables-service サービスを開始します。

```
$ sudo systemctl start iptables
$ systemctl status iptables
iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; vendor
preset: disabled)
   Active: active (exited) since Tue 2017-12-05 21:53:41 PST; 55s ago
  Process: 49042 ExecStart=/usr/libexec/iptables/iptables.init start
(code=exited, status=0/SUCCESS)
 Main PID: 49042 (code=exited, status=0/SUCCESS)

Dec 05 21:53:41 knc4-jf-intel-com systemd[1]: Starting IPv4 firewall with
iptables...
Dec 05 21:53:41 knc4-jf-intel-com iptables.init[49042]: iptables: Applying
firewall rules: [ OK ]
Dec 05 21:53:41 knc4-jf-intel-com systemd[1]: Started IPv4 firewall with
iptables.
```

ファイアウォール・ルールは、/etc/sysconfig/iptables ファイルで定義されます。

```
$ sudo cat /etc/sysconfig/iptables

# sample configuration for iptables service
# you can edit this manually or use system-config-firewall
# please do not ask us to add additional ports/services to this default
configuration
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
```

現在定義されているルールを確認します。何も定義されていないはずです。特定のマシンからパケットを受け付けるルールを追加するには、そのマシンの IP アドレスを指定します。

```
$ firewall-cmd --direct --get-all-rules
$ sudo firewall-cmd --direct --add-rule ipv4 filter INPUT 0 -s 10.23.3.61 -j
ACCEPT
success
$ firewall-cmd --direct --get-all-rules
ipv4 filter INPUT 0 -s 10.23.3.61 -j ACCEPT
```

新しいルールを追加したら、上記のコマンドラインを実行して、新しいルールが追加されたことを確認します。MPI プログラムを再度実行して、動作することを確認します。

```
$ mpirun -host localhost -n 1 /opt/intel/impi/2018.0.128/bin64/IMB-MPI1
Sendrecv : -host 10.23.3.61 -n 1  /opt/intel/impi/2018.0.128/bin64/IMB-MPI1
#------------------------------------------------------------
#    Intel (R) MPI Benchmarks 2018, MPI-1 part
#------------------------------------------------------------
# Date                  : Tue Dec  5 21:58:20 2017
# Machine               : x86_64
# System                : Linux
# Release               : 3.10.0-327.el7.x86_64
# Version               : #1 SMP Thu Nov 19 22:10:57 UTC 2015
# MPI Version           : 3.1
# MPI Thread Environment:

# Calling sequence was:

# /opt/intel/impi/2018.0.128/bin64/IMB-MPI1 Sendrecv

# Minimum message length in bytes:   0
# Maximum message length in bytes:   4194304
#
# MPI_Datatype                   :   MPI_BYTE
# MPI_Datatype for reductions    :   MPI_FLOAT
# MPI_Op                         :   MPI_SUM
#
#

# List of Benchmarks to run:
# Sendrecv

#-----------------------------------------------------------------------------
# Benchmarking Sendrecv
# #processes = 2
#-----------------------------------------------------------------------------
    #bytes #repetitions  t_min[usec]  t_max[usec]  t_avg[usec]   Mbytes/sec
         0         1000        16.49        16.49        16.49         0.00
         1         1000        16.40        16.40        16.40         0.12
         2         1000        16.40        16.40        16.40         0.24
         4         1000        16.86        16.86        16.86         0.47
         8         1000        16.43        16.43        16.43         0.97
        16         1000        16.32        16.32        16.32         1.96
        32         1000        16.64        16.64        16.64         3.85
        64         1000        16.90        16.90        16.90         7.57
       128         1000        16.86        16.86        16.86        15.18
       256         1000        29.58        29.60        29.59        17.30
       512         1000        27.73        27.74        27.74        36.91
      1024         1000        28.07        28.09        28.08        72.91
      2048         1000        34.95        34.97        34.96       117.15
      4096         1000        36.22        36.23        36.22       226.12

^C[mpiexec@knc4] Sending Ctrl-C to processes as requested
[mpiexec@knc4] Press Ctrl-C again to force abort
```

このルールを削除するには、次のコマンドを実行します。

```
$ sudo firewall-cmd --direct --remove-rule ipv4 filter INPUT 0 -s 10.23.3.61 -j
ACCEPT
success
$ firewall-cmd --direct --get-all-rules
```

まとめ

ファイアウォールによってノード間の MPI 通信がブロックされることがあります。この記事では、MPI ランク間の通信を可能にする 3 つの手法を紹介しました。

コンパイラーの最適化に関する詳細は、最適化に関する注意事項を参照してください。