Oracle Connection Managerを使用したプロトコルの切り替え

Oracleホワイト・ペーパー / 2019年12月5日

ORACLE

本書の目的

Oracle Connection Manager（Oracle CMAN）には、プロトコルを切り替えるためのさまざまな方法が用意されているため、Oracle CMANを使用すれば、両側で異なる送信用プロトコルを使用したクライアントとサーバー間の接続が実現します。このホワイト・ペーパーでは、Oracle CMANがさまざまなプロトコルを使用してOracle Netトラフィックをルーティングする際に用いる複数のアプローチの概要を説明します。Oracle CMANのユースケース、例、およびNEXT_HOP機能の詳しい構成についても紹介します。

免責事項

目次

概要

Oracle Connection Manager（Oracle CMAN）は、クライアント接続リクエストが次のホップにルーティングされる際に経由する透過的なプロキシです。別のOracle CMAN、SCANリスナー、またはデータベース・リスナーが次のホップになり得ます。Oracle CMANにより、あらゆるRACクラスタをはじめとするデータベース・サブセットがクライアントから見えなくなります。Oracle CMANの背後には複数のデータベースをデプロイでき、クライアントにデータベースへの単一のエントリ・ポイントが提供されます。クライアントは、セッションの多重化、圧縮、TLSセキュリティ、アクセス制御など、Oracle CMANで構成される機能を透過的に活用できます。

Oracle CMANは、着信接続要求を処理しながら、プロトコルの切り替えも実行します。着信と発信の各接続で異なるプロトコルが使用される場合、プロトコルの切り替えが必要です。たとえば、Oracle CMANでは、TCPS（TLSを使用したTCP）経由で着信クライアント接続を受信し、発信接続でTCPを使用すること（またその逆）が可能です。Oracle CMANによるプロトコルの切り替えは、クライアントとサーバーの双方に対して透過的に行われます。つまり、この機能を使用するためにアプリケーションに変更を加える必要はありません。

Oracle CMANは以下の3つの方法で、着信接続要求を次のホップ・ネットワーク・アドレスにプロキシすることができます。

- 動的なサービス登録：remote_listenerデータベース・パラメータを使用して、データベースがOracle CMANにそのサービスとプロトコル・アドレスを動的に登録します。Oracle CMANはサービスのクライアント・リクエストを受信すると、登録されているサービス情報を基に、その接続を適切なサービス・ハンドラに転送します。
- SOURCE_ROUTE：クライアントは、source_routeを指定し、Oracle CMANへの接続時に次のホップ・プロトコル・アドレスを接続文字列で提供できます。Oracle CMANはこのアドレスを使用してこの接続を転送します。
- NEXT_HOP：Oracle CMANの構成において、次のホップ・アドレスを固定できます。Oracle CMANはこのアドレスを使用してすべての着信接続要求を転送します。

ユースケース

プロトコルの切り替えは、次のようなさまざまなデプロイメント・シナリオにおいて有用です。

1. 最新のTLSバージョンをサポートしていない古いクライアント。このようなクライアントは、Oracle CMANインスタンスを介してデータベース・サーバーに接続をルーティングすることで、Oracle CMANがサポートする新しいTLSバージョンを活用できます。その際、Oracle CMANはクライアントとサーバー間の通信を可能にするために、TCPとTCPS（TCP経由のTLS）間でプロトコルの切り替えを実行します。
2. プロトコルの切り替えは、片側がセキュアなネットワーク内にあり、もう片側がセキュアでないネットワーク内にあるようなデプロイメントにおいても有用です。クライアントまたはサーバーのいずれかで非セキュアなプロトコルを使用するという柔軟な選択が可能になり、その結果、暗号化と復号化のオーバーヘッドがOracle CMANにオフロードされます。
3. Oracle CMANでNEXT_HOPを使用する際に、データベースでリモート・サービスの登録を行う必要はありません。プロトコルの切り替えは、さまざまなクライアントで単一のアウトバウンド・ポイントが必要なデプロイメントにおいても有用です。
4. Oracle CMANは、IPv4およびIPv6ネットワーク間のブリッジとして使用できます。

プロトコルの切り替えでのNEXT_HOPの使用

プロトコルを切り替え、クライアント・リクエストを転送する方法の1つは、固定された次のホップ・プロトコル・アドレスをOracle CMANで構成することです。NEXT_HOPはバージョン18c以降でサポートされます。

例

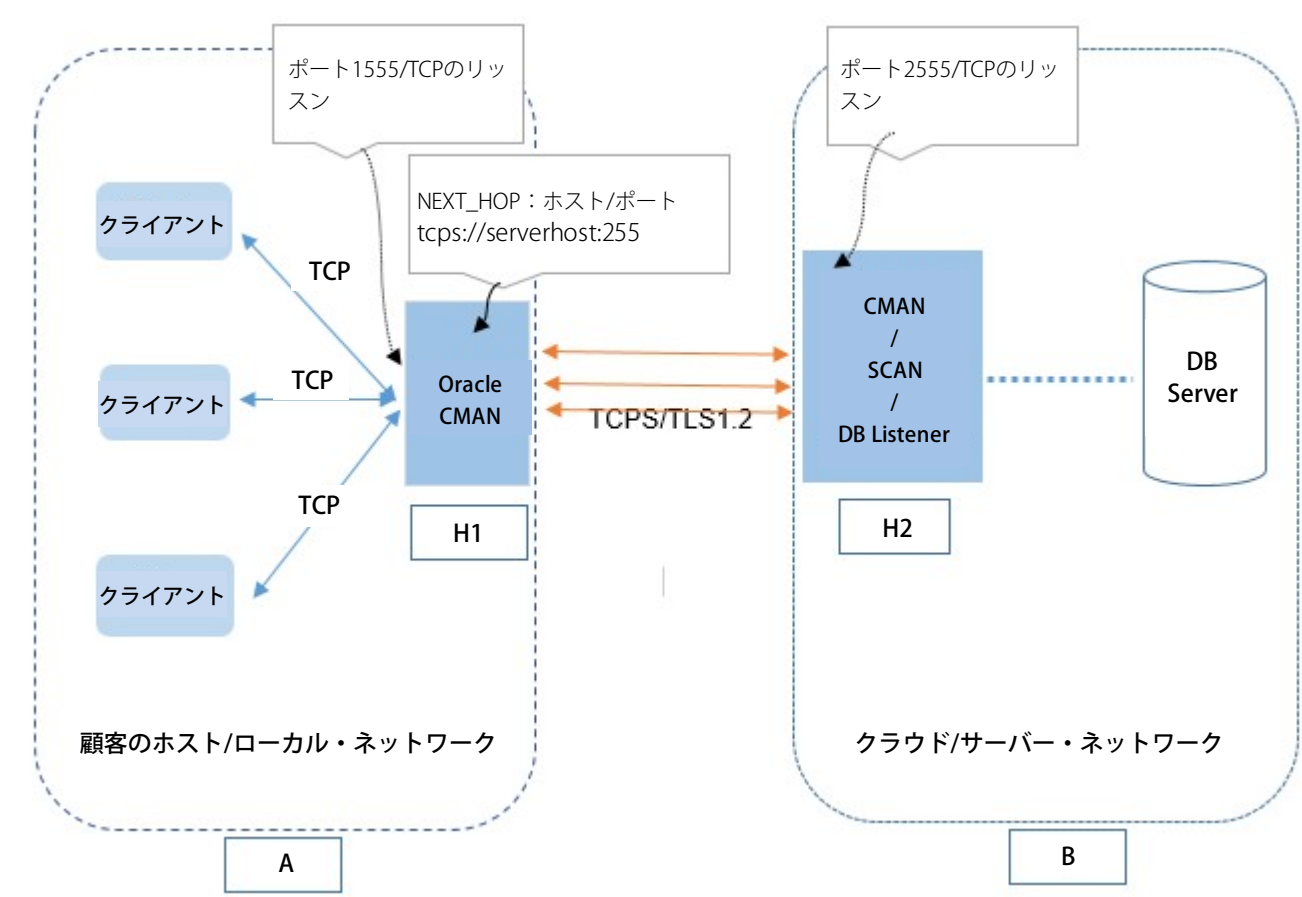

図1：データベースへの接続でTCPからTCPSへのプロトコル変換を使用するクライアント向けのOracle CMANのデプロイメント

図1に示すデプロイメントでは、クライアントは左側のボックスA、サーバーはボックスBに表示されています。クライアントは、ライブラリが最新のTLSバージョン（1.2）をサポートしていない古いOracleバージョンを使用しています。Oracle CMAN（バージョン18c以降）がボックスAにデプロイされ、アプリケーション・サーバー・ホストまたはクライアント・サブネットのいずれかの役割を果たしています。Oracle CMANの正確な配置は、セキュリティ要件によって異なります。このOracle CMANは、TCPアドレス1555でリッスンします。また、NEXT_HOPがサーバー側リスニング・エンドポイント（ポート2555/TCPS）に構成されています。このOracle CMANではTLS1.2がサポートされます。

この例では、H2がSCANリスナーまたはデータベース・リスナーの場合、クライアントとサーバー間の接続は次のようになります。

クライアント **--[TCP]-->** Oracle CMAN **--[TCPS]-->** データベース

H2は別のOracle CMANにもなり得ます。その場合、クライアントとサーバー間の接続はおそらく次のようになります。

クライアント --[TCP]--> Oracle CMAN（H1） --[TCPS]--> Oracle CMAN（H2）--TCP--> データベース

ここで、双方のOracle CMANはTCPとTCPS間の切り替えプロトコルです。

プロトコルの切り替えは、TCPやTCPS、もしくは上記の例に限定されません。サポートされる任意のSQL*Netプロトコルを使用できます。

cman.oraでのNEXT_HOPの構成

NEXT_HOPパラメータは、cman.ora内の構成の下の別のセクションで指定します。

NEXT_HOPを構成するcman.oraのサンプルを以下に示します。

```
CMAN=
      (CONFIGURATION=
          (ADDRESS=(PROTOCOL=tcp)(HOST=proxysvr)(PORT=1555))
          (rule_list=
              (rule=(src=*)(dst=*)(srv=*)(act=accept))
           )
          (PARAMETER_LIST=
              (MAX_GATEWAY_PROCESSES=8)
              (MIN_GATEWAY_PRCESSSES=3)
           )
          (NEXT_HOP=(ADDRESS=(PROTOCOL=tcps)(HOST=serverhost)(PORT=2555)))
       )
WALLET_LOCATION= (source=(method=file)(method_data= (directory=$WALLET_LOC_PATH)))
```

注：上記の例では、proxysvrの代わりにlocalhostを使用することで、ローカル・ノードへの着信接続要求を制限できます。そうすることで、リモート・クライアントはこのOracle CMANに接続してデータベース・サーバーにルーティングすることができなくなります。これにより、非暗号化プロトコルはクライアント・ネットワーク内であっても有線接続されなくなります。

NEXT_HOPパラメータの値は、次のサーバーのプロトコル・アドレスです。クライアント接続リクエストが受信されるたびに、Oracle CMANはこのアドレスに接続する必要があります。

NEXT_HOPの可能な形式と値は以下のとおりです。

a) 単一アドレス

```
(NEXT_HOP=(ADDRESS=(PROTOCOL=tcps)(HOST=serverhost)(PORT=2555)))
```

b) 複数アドレス：ADDRESS_LISTやDESCRIPTIONを使用して、load_balance、failoverといった他の特性とともに複数のアドレスを指定できます。

```
(NEXT_HOP=(ADDRESS_LIST=
                (LOAD_BALANCE=on)
```

```
                    (FAILOVER =on)
                    (ADDRESS=(PROTOCOL=tcps)(HOST=serverhost1)(PORT=2555))
                    (ADDRESS=(PROTOCOL=tcp)(HOST=serverhost1)(PORT=2556)))
```

または

```
(NEXT_HOP=(DESCRIPTION=
                    (LOAD_BALANCE=on)
                    (FAILOVER=on)
                    (ADDRESS=(PROTOCOL=tcps)(HOST=serverhost1)(PORT=2555))
                    (ADDRESS=(PROTOCOL=tcps)(HOST=serverhost1)(PORT=2557)))
```

注：

- 複数のアドレスが指定されている場合、フェイルオーバーはデフォルトで有効になります。
- TCPS（TLSを使用したTCP）プロトコルが次のホップ・プロトコル・アドレスに使用される場合、cman.oraでwallet_locationを指定する必要があります。以下に例を示します。

```
WALLET_LOCATION= (source=(method=file)(method_data= (directory=$WALLET_LOC_PATH)))
```

- NEXT_HOPはリロードが可能なパラメータです。この値をcman.oraで変更した後に、Oracle CMANを再起動する必要はありません。パラメータのリロードは、cmctl **reload**コマンドを使用して行うことができます。

Oracle CMAN経由で確立されたクライアント接続の検証

Linuxでは、netstatコマンドを使用して、Oracle CMAN経由で確立された接続を検証できます。以下は、SQL*Plusを使用して、ポート1522のローカル・マシンでリッスンするOracle CMANに接続している場合の例です。

```
$ netstat –anp | grep sqlplus
tcp        0      0 10.90.137.2:9992        10.90.137.2:1522        ESTABLISHED 13015/sqlplus
```

次に、クライアント・ポート9992を使用するプロセスを特定します。

```
$ netstat -anp | grep 9992
tcp        0      0 10.90.137.2:9992        10.90.137.2:1522        ESTABLISHED 13015/sqlplus
tcp6       0      0 10.90.137.2:1522        10.90.137.2:9992        ESTABLISHED 11796/cmgw
```

注：2行目は、Oracle CMANのゲートウェイ・プロセス（cmgw）のピア情報です。

次に、このゲートウェイ・プロセスの接続を表示します。

```
$ netstat -anp | grep 11796
tcp        0      0 10.90.137.2:43305       10.90.137.2:1521        ESTABLISHED 11796/cmgw
tcp6       0      0 10.90.137.2:1522        10.90.137.2:9992        ESTABLISHED 11796/cmgw
```

注：1行目は、このゲートウェイ・プロセスがDBリスナー・ポート1521に接続されていることを示しています。

さらに、接続を確立中のOracle CMANログ・ファイルとリスナー・ログ・ファイルの双方の末尾に、新しいログ・エントリが表示されています。事後分析において、タイムスタンプを使用してログ・エントリを照合することができます。たとえば、上記のSQL*Plus接続の場合、次のような内容がOracle CMANのログに記録されていることが予想されます。

26-SEP-2019 13:03:52 *

(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)(CID=(PROGRAM=sqlplus)(HOST=myhost)(USER=self))) *
(ADDRESS=(PROTOCOL=tcp)(HOST=10.90.137.2)(PORT=9992)) * establish * sales.us.example.com * 0

同様に、リスナー・ログには次のような内容が記録されていることが予想されます。

26-SEP-2019 13:03:52 *

(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)(CID=(PROGRAM=sqlplus)(HOST=myhost)(USER=self))) *
(ADDRESS=(PROTOCOL=tcp)(HOST=10.90.137.2)(PORT=43305)) * establish * sales.us.example.com * 0

Oracle CMANログ・エントリのソース・ポート（9992）がnetstatによって表示されたSQL*Plusのポートと同一であり、リスナー・ログ・エントリのソース・ポート（43305）がOracle CMANゲートウェイ・プロセスのポートと同一であることに気付きます。

確立された接続の数は、cmctl show connectionsコマンドを使用して監視できます。このコマンドを使用するには、Oracle CMANでパラメータconnection_statisticsがyesに設定されている必要があります。このコマンドにより、接続数、または各接続の詳細のいずれかが表示されます。以下に例を示します。

Connection ID		512
Gateway ID		1
Source		10.90.137.2
Destination		10.90.137.2
Service		sales.us.example.com
State		ESTABLISHED
Idle time		15
Connected time		15
Bytes Received	[IN]	2783
Bytes Received	[OUT]	4277
Bytes Sent	[IN]	4277
Bytes Sent	[OUT]	2783

結論

Oracle CMANのプロトコルの切り替え機能は、オンプレミスとクラウドを含むさまざまなデプロイメント・シナリオで使用できます。Oracle CMANのNEXT_HOP構成により、あらゆる着信リクエストを特定の宛先にルーティングする方法が提供されます。設定は非常にシンプルで、複雑な登録は不要です。この機能を使用すると、互換性のある古いクライアントが、古いクライアントでサポートされないセキュアなプロトコルを使用して、新しいサーバーと通信できるようになります。

ORACLE CORPORATION

Worldwide Headquarters

500 Oracle Parkway, Redwood Shores, CA 94065 USA

海外からのお問い合わせ窓口

電話　+ 1.650.506.7000+ 1.800.ORACLE1

FAX　+ 1.650.506.7200

oracle.com

オラクルの情報を発信しています

+1.800.ORACLE1までご連絡いただくか、oracle.comをご覧ください。北米以外の地域では、oracle.com/contactで最寄りの営業所をご確認いただけます。

blogs.oracle.com/oracle　facebook.com/oracle　twitter.com/oracle

Integrated Cloud Applications & Platform Services

ホワイト・ペーパー **Oracle Connection Managerを使用したプロトコルの切り替え**Oracle Connection Managerを使用したプロトコルの切り替えOracle Connection Managerを使用したプロトコルの切り替え
2019年12月

ORACLE