



Oracle ホワイト・ペーパー
2013年6月

Oracle Single Client Access Name (SCAN)

はじめに	1
SCANを使用するためのネットワーク要件	2
オプション1 – 自社DNSを使用する	2
オプション2 – グリッド・ネーミング・サービス (GNS) を使用する	4
インストール時にDNSサーバーが利用できない場合の代替方法	4
Oracle Grid Infrastructure 11g Release 2によるSCANの構成	5
Oracle Grid Infrastructure 12c Release 1によるSCANの構成	6
SCANに関する複数のサブネットのサポートの有効化	7
SCANを使用したOracle Database構成	10
SCANを使用したクライアントのロードバランシング	10
複数のサブネットのサポートとLISTENER_NETWORKS	11
バージョンおよび下位互換性	12
その他のSCAN関連構成	13
同じサブネット上の複数のポートによるSCANの使用	13
GDSを使用しないMAA環境でのSCANの使用	14
GDSを使用するMAA環境でのSCANの使用	14
Oracle Connection ManagerによるSCANの使用	15
まとめと結論	15

はじめに

Single Client Access Name (SCAN) は、Oracle Real Application Clusters環境で使用される機能です。SCANを使用すると、クラスタ内で実行中のOracle Databaseにアクセスする際に、クライアントが単一の名前を使用できます。SCANは、クラスタ内のデータベースのクラスタ・エイリアスと考えることができます。この機能のメリットは、クラスタにノードやデータベースを追加したり削除したりしても、SCANを使用しているクライアントの接続情報を変更する必要がないという点です。

SCANはOracle Real Application Clusters (Oracle RAC) 11g Release 2で初めて導入され、Oracle RAC 12cで機能が追加されました。クラスタにアクセスしてそのクラスタ内のデータベースに接続するときに使用する名前が1つであるため、クライアントはEZConnectとシンプルなJDBC Thin URLを使用して、クラスタ内で稼働中のデータベースにアクセスできます。クラスタ内で動作しているデータベースまたはサーバーの数を意識する必要はなく、リクエストされたデータベースが実際にクラスタ内のどのサーバーで動作していても関係ありません。

EZconnect	<code>sqlplus system/manager@sales1-scan:1521/oltp</code>
JDBC接続	<code>jdbc:oracle:thin:@sales1-scan:1521/oltp</code>

例1 : EZConnectとThin JDBCの接続文字列の例

SCANを使用するためのネットワーク要件

デフォルトのSCAN構成は、Oracle Database 11g Release 2以上とともに提供されるOracle Grid Infrastructureのインストール時に定義されます。Oracle Grid Infrastructureは、Oracle ClusterwareとOracle Automatic Storage Managementを含む単一のOracleホームです。

Oracle RAC 11g Release 2以上を使用するには、Oracle Grid Infrastructureを先にインストールする必要があります。Oracle Grid Infrastructureのインストール時におけるやり取りの中で、SCAN名の入力を要求されます。SCANを定義するには、2つのオプションがあります。

1. 自社DNS（ドメイン・ネーム・サービス）を使用してSCANを定義する
2. グリッド・ネーミング・サービス（GNS）を使用してSCANを定義する

オプション1 – 自社DNSを使用する

オプション1を選択した場合は、ネットワーク管理者に依頼して、ラウンドロビン方式で3つのIPアドレスに解決される単一の名前を1つ以上作成します。ロードバランシングと可用性の要件を考慮すると、IPアドレスは、クラスタに含まれるサーバーの数に関係なく、3つにすることが推奨されます。

3つのIPアドレスは、クラスタ内のデフォルトのパブリック・ネットワークと同一サブネット上に存在していなければなりません。名前の長さは15文字以下にします。ドメインは含まれません。また、ドメイン接尾辞なしで解決できる必要があります（例："scan1-scan.example.com"ではなく、"sales1-scan"で解決できる必要があります）。ネットワーク・インタフェースへのIPの割当てはOracle Clusterwareによって行われるため、ここにはIPを割り当てないようにしてください。

sales1-scan.example.com	IN A 133.22.67.194
	IN A 133.22.67.193
	IN A 133.22.67.192

例2：SCANのDNSエントリの例

DNSのSCAN構成は、"nslookup"を使用して確認できます。SCANエントリによって解決されたIPにラウンドロビン方式でアクセスするようにDNSを設定している場合は、"nslookup"コマンドを2回以上実行して、ラウンドロビン・アルゴリズムが機能していることを確認します。"nslookup"を実行するたびに3つのIPが異なる順序で返されれば、アルゴリズムは正常に機能しています。

1回目のnslookup	2回目のnslookup
<pre>[oracle@mynode] nslookup sales1-scan Server: 131.32.249.41 Address: 131.32.249.41#53 Non-authoritative answer: Name: sales1-scan.example.com Address: 133.22.67.192 Name: sales1-scan.example.com Address: 133.22.67.193 Name: sales1-scan.example.com Address: 133.22.67.194</pre>	<pre>[oracle@mynode] nslookup sales1-scan Server: 131.32.249.41 Address: 131.32.249.41#53 Non-authoritative answer: Name: sales1-scan.example.com Address: 133.22.67.193 Name: sales1-scan.example.com Address: 133.22.67.194 Name: sales1-scan.example.com Address: 133.22.67.192</pre>

例3: "nslookup"によるDNSのSCAN構成の参照

注: 例3のように3つのIPのセットがDNSサーバーから返されない場合、またはDNSサーバーがラウンドロビン方式でない場合は、ネットワーク管理者に依頼して、ラウンドロビンを正しく設定してください。DNSレベルでのラウンドロビンにより、クラスタ内を浮動するSCANリスナー全体で接続リクエストのロードバランシングが可能になります。SCAN全体が機能している必要はなく、そのように設定されていなくても、リストの最初のSCANリスナーが停止すると、別のSCANリスナーへの接続リクエストのフェイルオーバーが行われます。

クラスタ内のSCANリスナー間での接続リクエストのフェイルオーバーは通常、Oracleクライアントによって実行されます。Oracle Database 11g Release 2以上のバージョンのOracleクライアントでは、このタイプのフェイルオーバーを実行するために特別な構成は必要ありません。古いバージョンのクライアントでは、追加の構成を考慮する必要があります¹。そのため、SCANを使用してデータベースに接続する際は、Oracle Database 11g Release 2以上のバージョンのクライアントを使用することを推奨します。

クライアント側のDNSキャッシュを使用することにより、DNSラウンドロビンがDNSサーバーから発生しない（例3のように3つのIPのセットがDNSから返されない）という誤解が生じる可能性があります。通常、クライアント側のDNSキャッシュは、外部DNSサーバーに送信されるDNSリクエストを最小限に抑え、DNS解決にかかる時間を最短にするために使用されます。これは、ローカル・アイテムによる単なる再帰的DNSサーバーです。

ラウンドロビンをローカルで提供するようにクライアント側のDNSを設定できない場合、またはクライアント側のDNSを無効にできない場合、JDBC Thin接続を使用するOracleクライアントは通常、リストで最初に返されるSCAN-IPおよびSCANリスナーへの接続を試みます。これによって基本的には、クラスタ内のSCANリスナー全体でクライアントからの接続リクエストのロードバランシングが無効になりますが、SCAN機能全体への影響はありません。Oracle Call Interface (OCI) ベースのデータベース・アクセス・ドライバが内部ラウンドロビン・アルゴリズムを適用するため、この場合は考慮する必要はありません。

¹ 詳細については、本書の「SCANを使用する場合のOracleクライアントとOracle Databaseのバージョンの互換性」を参照してください。

オプション2 – グリッド・ネーミング・サービス (GNS) を使用する

オプション2を選択した場合は、インストール時のやり取りの中でSCAN名を入力するだけで済みます。クラスタ構成のある段階で、3つのIPアドレスがDHCPサービスから取得されるか、Oracle RAC 12cでIPv6ベースのIPアドレスを使用している場合は"ステートレス・アドレス自動構成" (SLAAC) を使用して取得され (ただしGNSを使用すると、パブリック・ネットワーク上で何らかのIPアドレスの動的割当てを使用しているとみなされます)、SCANが作成されます。その後、SCANの名前解決がGNSによって実行されます²。

インストール時にDNSサーバーが利用できない場合の代替方法

クラスタ内にOracle RAC 11g Release 2以上のデータベースを作成する上で、SCANの概念は不可欠です。そのため、Oracle Grid Infrastructureのインストール時にデフォルトのSCAN解決を設定するようにOracle Universal Installerから要求されます。データベースを作成するときに使用するOracle Database 11g Release 2以上のツール (Oracle Database Configuration Assistant (Oracle DBCA)、Oracle Network Configuration Assistant (Oracle NetCA) など) はすべて、SCANが設定されていることを前提に動作します。したがって適切なSCAN解決を設定するまでは、Oracle Universal Installerの手順を続行できません。

ただし代替方法としてホスト・ファイル・ベースの解決を使用することで、DNSベースのSCAN解決を設定せずにインストール要件を満たすことができます。その場合は、代表的なホスト・ファイル・エントリを1つ使用して、SCANを1つのIPアドレス以外には解決しないようにします。ローカル・ホスト・ファイルを使用する場合は、DNSサーバーで実行されるラウンドロビン方式の解決をシミュレートすることはできません。OSによって実行されるホスト・ファイル参照では、名前に一致する最初のIPアドレスのみが返されます。エントリが1つ (ホスト・ファイルに1行ある) の場合も同様です。したがって、クラスタに対して1つのSCANのみ作成します (その場合は、クラスタ内のすべてのノード上のホスト・ファイルを変更する必要があります)。

この代替方法は、(Oracle Database 11g Release 2より前の) 旧リリースからのアップグレードを実行する場合にも使用できます。ただし、アップグレードまたは初期インストールの直後に、本書の"オプション1"または"オプション2"で説明したSCAN構成を有効にすることを強く推奨します。SCAN構成の変更をクラスタに認識させるには、ホスト・ファイル内のエントリを削除してから、クラスタ内の1つのノード上でrootユーザーとして"`srvctl modify scan -n <scan_name>`"を実行します。

scan_nameには既存の完全修飾名 (または新しい名前) を指定できますが、すでに説明したとおり、3つのIPが関連付けられていて、DNSを使用して解決できる名前を指定する必要があります。残りの再構成処理は自動的に実行されます。

² グリッド・ネーミング・サービスを使用してクラスタをインストールする方法についての詳細は、『Oracle® Grid Infrastructureインストール・ガイド11gリリース2 (11.2)』 (http://docs.oracle.com/cd/E16338_01/install.112/b57234/presolar.htm#CHDDBHGF) を参照してください。

Oracle Grid Infrastructure 11g Release 2によるSCANの構成

クラスタの構成時に、クラスタ内にSCAN用のリソースが作成されます。SCANの解決先となる3つのIPアドレスについて、それぞれSCAN VIPリソースとSCANリスナーが作成されます。SCANリスナーはSCAN VIPに依存しており、3つのSCAN VIP（および対応するリスナー）はクラスタ全体に分散配置されます。

つまり、クラスタが3つ以上のノードで構成されている場合、各リソース・ペア（SCAN VIPとリスナー）はクラスタ内の異なるノード上で起動します。

2ノード・クラスタを使用している場合（この場合も、単純化するという理由から、3つのIPを推奨します）は、通常運用時にはクラスタ内の1つのサーバーが2つのSCANリソース・セットをホストします。SCAN VIPを実行しているノード上で障害が発生すると、そのSCAN VIPと対応するリスナーがクラスタ内の別のノードにフェイルオーバーされます。こうした障害によりクラスタ内の使用可能なサーバー数が3より少なくなった場合も、1つのサーバーが2つのSCANリソース・セットをホストします。クラスタ内のノードが再度使用可能になると、先ほど説明した分散配置が有効になり、それを受けて1つのセットが別ノードへ移動します。

```
[grid@mynode] srvctl config scan_listener
SCAN Listener LISTENER_SCAN1 exists. Port: TCP:1521
SCAN Listener LISTENER_SCAN2 exists. Port: TCP:1521
SCAN Listener LISTENER_SCAN3 exists. Port: TCP:1521

[grid@mynode] srvctl config scan
SCAN name: sales1-scan, Network: 1/133.22.67.0/255.255.252.0/
SCAN VIP name: scan1, IP: /sales1-scan.example.com/133.22.67.192
SCAN VIP name: scan2, IP: /sales1-scan.example.com/133.22.67.193
SCAN VIP name: scan3, IP: /sales1-scan.example.com/133.22.67.194
```

例4：Oracle Grid Infrastructure 11g Release 2でのSCANの構成例

Oracle Grid Infrastructure 12c Release 1によるSCANの構成

Oracle Grid Infrastructure 11g Release 2に関して説明したSCAN設計の原則の大半は、Oracle Grid Infrastructure 12cにも当てはまります。ただし、ユーザーの要望とフィードバックに基づいて、Oracle Grid Infrastructure 12cではSCANの概念が次のように拡張されました。

1. SCANとOracle Clusterwareで管理されるVIPがIPv6ベースのIPアドレスをサポートするようになりました。
2. SCANは、デフォルトでは、クラスタ内のノードからのサービス登録だけを受け入れるように制限されています。
3. SCANは、クラスタ内の複数のサブネットをサポートします（サブネット当たり1つのSCAN）。

Oracle Grid InfrastructureのOracle Universal Installerベースのインストール中は、（デフォルトのネットワーク（通常、ネットワーク番号1）において）デフォルトのSCANだけをインストールして構成できます。クラスタ内の複数のサブネットのサポートは、インストール後に有効にする必要があります。

これらの拡張のために、SCANおよびSCAN_LISTENERの構成を変更する必要があります。

```
[grid@mynode]$ srvctl config scan
SCAN name: sales1-scan.example.com, Network: 1
Subnet IPv4: 133.22.67.0/255.255.252.0/eth0
Subnet IPv6:
SCAN 0 IPv4 VIP: 133.22.67.194
SCAN name: sales1-scan.example.com, Network: 1
Subnet IPv4: 133.22.67.0/255.255.252.0/eth0
Subnet IPv6:
SCAN 1 IPv4 VIP: 133.22.67.193
SCAN name: sales1-scan.example.com, Network: 1
Subnet IPv4: 133.22.67.0/255.255.252.0/eth0
Subnet IPv6:
SCAN 2 IPv4 VIP: 133.22.67.192

[grid@mynode]$ srvctl config scan_listener
SCAN Listener LISTENER_SCAN1 exists. Port: TCP:1521
Registration invited nodes:
Registration invited subnets:
SCAN Listener LISTENER_SCAN2 exists. Port: TCP:1521
Registration invited nodes:
Registration invited subnets:
SCAN Listener LISTENER_SCAN3 exists. Port: TCP:1521
Registration invited nodes:
Registration invited subnets:
```

例5：SRVCTLの出力の例

例5に示されている出力の例からも分かるように、SCANに関するIPv6ベースのIPのサポート（また、それに伴うノードVIPのサポート）は、Oracle Grid Infrastructure 12cのSCANの基本概念になっています。例5では、IPv6ベースのIPアドレスは使用されておらず、SCANが1つだけクラスタ内のネットワーク番号1上にデプロイされ、ネットワーク・インタフェース・カード（NIC）eth0に割り当てられています。

例5の出力の例には、SCANの新しいセキュリティ機能のトレースも示されています。デフォルトでは、Oracle Grid Infrastructure 12cのSCANは、SCANリスナーが割り当てられているクラスタに含まれるデータベース・インスタンスの登録だけを（リスナーへの動的データベース・インスタンス登録の一部として）受け入れます。

注：クラスタでホストされるデータベースのデータベース・インスタンス登録のみを許可するために、割り当てられているSCANによるデータベース・インスタンスのデフォルトの登録では、プライベート・インターコネクトが使用されます。何らかの理由で、そのクラスタ用に定義されているプライベート・インターコネクトにアクセスできない（またはそれを使用しない）データベースをそれぞれのSCANリスナーで登録する必要がある場合は、次の例6に示されているように、（指定された）ノードまたはサブネットごとに追加の登録を許可するようにSCANリスナーを変更できます。招待されるノードは、将来参加する可能性のあるノードの事前承認のためにリストに追加される際には、クラスタ内に存在している必要がないことに注意してください。

```
[grid@mynode]$ srvctl modify scan_listener -invitednodes test -update
[grid@mynode]$ srvctl config scan_listener
SCAN Listener LISTENER_SCAN1 exists. Port: TCP:1521
Registration invited nodes: test
Registration invited subnets:
SCAN Listener LISTENER_SCAN2 exists. Port: TCP:1521
Registration invited nodes: test
Registration invited subnets:
SCAN Listener LISTENER_SCAN3 exists. Port: TCP:1521
Registration invited nodes: test
Registration invited subnets:
```

例6：追加ノードからの登録を許可するためのSCANリスナーの変更

SCANに関する複数のサブネットのサポートの有効化

SCANをクラスタ内の複数のサブネットで使用するには、まずクラスタでの複数のサブネットの使用を有効にする必要があります。この作業は通常、インストール後に次の手順で行われます。

1. （パブリック）ネットワークで追加のサブネットを作成します。
2. 新しく作成したサブネットにノードVIPを割り当てます。
3. 新しく作成したサブネット用のノード・リスナーを作成します。
4. オプション1または2を使用してSCANを作成します。

手順1では、Oracle RACクラスタでのパブリック・ネットワーク通信に使用する予定の複数のネットワーク・インタフェース・カード（NIC）があること、または別々のネットワークから複数のIPアドレスで使用する1つのインタフェース（またはインタフェースのセット1つ）があることが前提となっています。両方ともさまざまな方法で確立できますが、どちらの場合も例7のように、oifcfgコマンドによって指定されるとおりにOracle Clusterwareレイヤーが複数のインタフェースを認識できる必要があります。

```
[grid@mynode]$ oifcfg iflist
eth0 133.22.67.0
eth1 192.168.122.0
eth1 169.254.0.0
eth2 10.1.1.0

[grid@mynode]$ oifcfg setif -global eth2/10.1.1.0:public
[grid@mynode]$ oifcfg getif
eth0 133.22.67.0 global public
eth1 192.168.122.0 global cluster_interconnect,asm
eth2 10.1.1.0 global public

[root@mynode]# srvctl add network -netnum 2
-subnet 10.1.1.0/255.255.252.0/eth2
[root@mynode]# srvctl config network -netnum 2
Network 2 exists
Subnet IPv4: 10.1.1.0/255.255.252.0/eth0, static
Subnet IPv6:
```

例7：パブリック・ネットワークでの追加のサブネットの作成

上記のリストの手順2～4では、図8に示されているように、Oracle Clusterwareの標準コマンドを使用します。

```
#ADD node VIPs
[root@mynode]# srvctl add vip -node SalesSrv145 -netnum 2 -address SalesSrv
145v2/255.255.252.0
[root@mynode]# srvctl add vip -node SalesSrv146 -netnum 2 -address SalesSrv
146v2/255.255.252.0
[root@mynode]# srvctl add vip -node SalesSrv147 -netnum 2 -address SalesSrv
147v2/255.255.252.0
[root@mynode]# srvctl add vip -node SalesSrv148 -netnum 2 -address SalesSrv
148v2/255.255.252.0

#ADD node listener on network number 2
[grid@mynode]$ srvctl add listener -listener ListNet2 -netnum 2 -endpoints "TCP:1528"

#ADD SCAN on network number 2
[root@mynode]# srvctl add scan -scanname scantest -netnum 2

#START node VIPs
[root@mynode]# srvctl start vip -vip SalesSrv145v2
[root@mynode]# srvctl start vip -vip SalesSrv146v2
[root@mynode]# srvctl start vip -vip SalesSrv147v2
[root@mynode]# srvctl start vip -vip SalesSrv148v2

#START ListNet2 node listener on network number 2
[grid@mynode]$ srvctl start listener -listener ListNet2
[grid@mynode]$ srvctl status listener -listener ListNet2
Listener LISTNET2 is enabled
Listener LISTNET2 is running on node(s): SalesSrv145, SalesSrv146, SalesSrv147 ...

#START SCAN on network number 2
[root@mynode]# srvctl start scan -netnum 2

#Check configuration and status for SCAN
[root@mynode]# srvctl config scan -netnum 2
SCAN name: scantest, Network: 2
Subnet IPv4: 10.1.1.0/255.255.252.0/eth2
Subnet IPv6:
SCAN 0 IPv4 VIP: 10.1.1.250

[root@mynode]# srvctl status scan -netnum 2
SCAN VIP scan1_net2 is enabled
SCAN VIP scan1_net2 is running on node SalesSrv146
```

例8：新しく作成されたサブネットでのノードVIPの割当てとSCANの作成

例7および例8については次の点に注意してください。

1. "[root@mynode]"によって導入される操作は、権限のあるユーザーとして実行される必要があります。
 - "[root@mynode]"によって導入される操作は、権限のあるユーザーとして実行できます。
2. "srvctl add vip"コマンドに含まれる"-address"の名前は、たとえば次のように、DNSまたは/etc/hostsによって有効なIPアドレスに解決されることが前提となっています。
 - SalesSrv 145v2は、10.1.1.0サブネットの有効な未使用のIPアドレスに解決される必要があります。
3. SCANの"scantest.example.com"は、オプション1またはオプション2を使用して作成されていません。
 - 代わりに、"インストール時にDNSサーバーが利用できない場合の代替方法"が使用されています。

またネットワーク番号2にSCANを追加しても、このネットワークではSCANリスナーが自動的に作成されないことに注意してください。ネットワーク番号2でSCANを最大限活用するには、このネットワークでSCANリスナーを作成して起動する必要があります。

```
#ADD SCAN_LISTENER to network number 2
[grid@mynode]$ srvctl add scan_listener -netnum 2 -listener testscan
-endpoints TCP:1528

[grid@mynode]$ srvctl config scan_listener -netnum 2
SCAN Listener TESTSCAN_SCAN1_NET2 exists. Port: TCP:1528
Registration invited nodes:
Registration invited subnets:

[grid@mynode]$ srvctl start scan_listener -netnum 2

[grid@mynode]$ srvctl status scan_listener -netnum 2
SCAN Listener TESTSCAN_SCAN1_NET2 is enabled
SCAN listener TESTSCAN_SCAN1_NET2 is running on node SalesSrv146
```

例9：ネットワーク番号2に新しく追加されたSCAN構成へのSCANリスナーの追加

パブリック通信用に異なるサブネットを使用して別のSCAN設定を行い、それぞれのノードVIP、ノード・リスナー、およびSCANリスナーを作成したら、このSCANを使用してそれらを登録する方法をデータベースに通知する必要があります。リスナー（特にSCANリスナー）によるデータベース・インスタンス登録の原則は、Oracle Database 12cの場合も、Oracle Database 11g Release 2と同じです。そのためこの原則については、本書の「複数のサブネットのサポートとLISTENER_NETWORKS」で一般的に説明します。

SCANを使用したOracle Database構成

Oracle Database 11g Release 2以上のOracle RACデータベース構成にはSCANが不可欠です。そのため、標準のOracleツール（前述のOracle DBCAなど）を使用してデータベースを作成する場合は、REMOTE_LISTENERパラメータにはデフォルトでSCANが設定されます。これにより、各インスタンスがリモート・リスナーとしてSCANリスナーに登録され、そのインスタンスで提供中のサービス、現在の負荷、そのインスタンスで処理すべき着信接続リクエストの数（推奨値）に関する情報が提供されます。

このような場合には、LOCAL_LISTENERパラメータを考慮する必要があります。LOCAL_LISTENERパラメータにはノードVIPを設定する必要があります。完全修飾ドメイン名が必要であれば、LOCAL_LISTENERには完全修飾ドメイン名（例：node-VIP.example.com）を設定します。デフォルトでは、クラスタ構成時にクラスタ内の各ノードにノード・リスナーが作成されます。クラスタで複数のサブネットを使用する場合は、初期インストールの後に追加のノードVIPを構成する必要があります（本書の「SCANに関する複数のサブネットのサポートの有効化」を参照）。

Oracle Grid Infrastructure 11g Release 2以上では、ノード・リスナーはOracle Grid Infrastructureホームから実行され、指定されたポート（デフォルトは1521）を使用してノードVIPをリスニングします。

以前のデータベース・バージョンとは異なり、アドレス・リスト・エントリでホストをSCAN（例：HOST=sales1-scan）に解決するサーバー側TNSNAMESのエイリアスをREMOTE_LISTENERパラメータに設定することは推奨されません。例10に示す簡略化された"SCAN:port"構文を使用してください。

NAME	TYPE	VALUE
local_listener	string	(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=133.22.67.111)(PORT=1521))))
remote_listener	string	sales1-scan.example.com:1521

例10：LOCALおよびREMOTE_LISTENERのデフォルト構成

注：簡易接続ネーミング・メソッドを使用する場合は、クライアント名の解決の参照に使用されるネーミング・メソッドの順序を指定するときにEZCONNECTを確実にリストに含めるために、SQLNET.ORAへの変更が必要になる場合があります（NAMES.DIRECTORY_PATH=(tnsnames, ldap, ezconnect)）。

SCANを使用したクライアントのロードバランシング

Oracle SQL*Net 11g Release 2を使用して接続しているクライアントの場合は、前述したとおり、DNSを使用してSCAN名を解決して3つのIPアドレスを取得します。クライアントはDNSから取得したリストを調査し、いずれかのIPアドレスによる接続を試みます。エラーを受信したクライアントは、別のアドレスを試してから、ユーザーまたはアプリケーションにエラーを返します。この動作は、前リリースでクライアント接続文字列にアドレス・リストを指定した場合における、クライアント接続のフェイルオーバーの動作原理と似ています。

SCANリスナーは接続リクエストを受信すると、要求されたサービスを提供しているインスタンスのうち、もっとも負荷の小さいものを確認します。そしてもっとも負荷の小さいインスタンスが稼働しているノード上のローカル・リスナーに、接続リクエストをリダイレクトします。その後、クライアントにローカル・リスナーのアドレスが割り当てられます。このローカル・リスナーが、最終的にデータベース・インスタンスに対する接続を作成します。

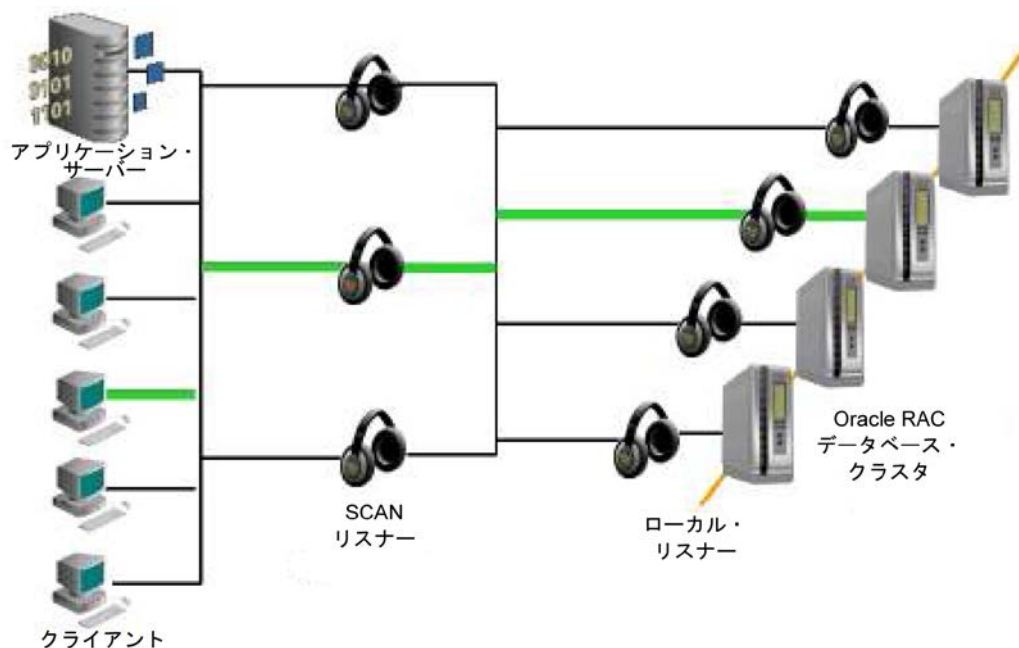


図1：SCANを使用した接続ロードバランシングのイメージ図

複数のサブネットのサポートとLISTENER_NETWORKS

Oracle 11g Release 2のドキュメント³には、LISTENER_NETWORKSパラメータに関して、「ネットワークには複数のローカル・リスナーとリモート・リスナーを含めることができます。デフォルトでは、すべてのリスナーが相互に登録されます。LISTENER_NETWORKS初期化パラメータに一連のリスナーを指定することで、リモート・リスナーのサブセットとともにローカル・リスナーのサブセットを指定できます。LISTENER_NETWORKSパラメータで指定されているリスナーを、LOCAL_LISTENERパラメータおよびREMOTE_LISTENERパラメータで指定しないでください」と説明されています。

複数のサブネットでSCANを使用する場合、LISTENER_NETWORKSパラメータは、Oracle Grid Infrastructure 11g Release 2によって異なるサブネットを介してアクセスされる可能性のあるすべてのデータベースに関して設定される必要があります。Oracle Grid Infrastructure 12cではOracle Database Agentが必要に応じて、LISTENER_NETWORKSパラメータをデフォルトで管理します。

³ LISTENER_NETWORKSパラメータの詳細については、Oracleのドキュメントを参照してください。

バージョンおよび下位互換性

クラスタ内のOracle RACデータベースへの接続にSCANを正しく使用できるかどうかは、クライアントがSCANを認識して使用できるかどうかだけでなく、前述のようにデータベースのREMOTE_LISTENERパラメータ設定が正しく構成されているかどうかにも左右されます。データベースに接続するOracleクライアントと、使用されるOracle Databaseのバージョンが、両方ともOracle Database 11g Release 2以上であり、本書で説明しているデフォルト構成を使用する場合は、通常はシステムに対する変更は不要です。

Oracleクライアントのバージョンと、クライアントが接続するOracle Databaseのバージョンが両方とも11g Release 2より前のバージョン（Oracle Database 11g Release 1、またはOracle Database 10g Release 2以前）の場合も同様です。その場合、11g Release 2より前のクライアントはTNS接続記述子を使用してクラスタのノードVIPを解決しますが、11g Release 2より前のOracleデータベースは引き続きREMOTE_LISTENERエントリを使用してノードVIPを参照します。

この構成のデメリットは、SCANが使用されないことです。そのため、バックエンドでクラスタが変更されるたびに、これまでと同様に、クライアントを変更する必要があります。Oracle Database 11g Release 2以上を使用している場合、クライアントが以前のバージョンのままであれば同じです。この問題を解決するには、OracleクライアントとOracle DatabaseのREMOTE_LISTENERの両方またはいずれかの設定をしかるべく変更します。

次のケースを検討しておく必要があります。

Oracleクライアントのバージョン	Oracle Databaseのバージョン	コメント
Oracle Database 11g Release 2以上	Oracle Database 11g Release 2以上	変更は不要です。
Oracle Database 11g Release 2以上	Oracle Database 11g Release 2より前	SCAN VIPをホストとしてREMOTE_LISTENERパラメータに追加します。
Oracle Database 11g Release 2より前	Oracle Database 11g Release 2以上	クライアントのTNSNAMES.oraを変更し、SCAN VIPを含めます（*下記を参照）。11g Release 2より前のデータベースからOracle DBUAを使用してアップグレードした場合は、REMOTE_LISTENERパラメータがノードVIPとSCANを参照するように自動的に構成されます。
Oracle Database 11g Release 2より前	Oracle Database 11g Release 2より前	SCANを使用できるようにする場合（推奨）は、次の作業を行います。SCAN VIPをホストとしてREMOTE_LISTENERパラメータに追加します。さらに、クライアントのTNSNAMES.oraを変更し、SCAN VIPを含めます（*下記を参照）。SCANを使用しない場合、変更は不要です。

表1：SCANを使用する場合のOracleクライアントとOracle Databaseのバージョンの互換性

注：11g Release 2より前のクライアント（Oracle Database 11g Release 1またはOracle Database 10g Release 2以前）を使用している場合は、SCANの利点を十分に活用できません。

その理由は、DNSから返されたSCAN用の3つのIPセットをOracleクライアントが処理できないためです。したがってクライアントは、リスト内の最初に返されたアドレスに対してのみ接続を試み、事実上、他のアドレスは無視します。その特定のIPをリスニングしているSCANリスナーを使用できない場合、またはIP自体を使用できない場合は、接続に失敗します。11g Release 2より前のクライアントでロードバランシングおよび接続のフェイルオーバーが実行されるようにするには、クライアントのTNSNAMES.oraを変更し、3つのアドレス行がクライアントで使用され、各アドレス行がいずれかのSCAN VIPに解決されるようにする必要があります。

```
sales.example.com =(DESCRIPTION=
  (ADDRESS_LIST= (LOAD_BALANCE=on) (FAILOVER=ON)
    (ADDRESS=(PROTOCOL=tcp)(HOST=133.22.67.192)(PORT=1521))
    (ADDRESS=(PROTOCOL=tcp)(HOST=133.22.67.193)(PORT=1521))
    (ADDRESS=(PROTOCOL=tcp)(HOST=133.22.67.194)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME= saleservice.example.com)))
```

例11 : Oracle Database 11g Release 2より前のクライアントで使用するTNSNAMES.ora

その他のSCAN関連構成

SCANは、複数の構成で使用できます。本書では、影響に注意すれば他の構成とともに使用できる推奨設定について説明します。以下の項では、比較的特殊な構成について説明します。これらの構成は今後、必要に応じて更新される予定です。

同じサブネット上の複数のポートによるSCANの使用

アプリケーションが、1つのSCANをホストとして使用してホスト/ポート・レベルで区別可能な接続文字列を保持できるように、またはポートベースのファイアウォール構成を有効にするために、同じサブネット上の複数のポートをサポートするようにSCANを設定できます。

異なるポートは、初期接続のためにのみ使用されることに注意する必要があります。SCANを使用して任意のSCANポートで接続が確立されると、ポートの割当ては保持されず、Oracle RACクラスタ内の接続のルーティングが、1つまたは複数のローカル・リスナーに登録されたデータベース・サービスを使用して実行されます。

このため、すべてのSCANが同様に処理され、どのSCANポートで確立された接続も、同じサブネット内の任意のポートで、任意の登録済みローカル・リスナーに送られる可能性があります。

また、複数ポートの割当てについてはバージョンによって異なることに注意してください。Oracle Grid Infrastructure 11.2.0.3では1つのSCANで複数のポートを設定する必要がなくなっていますが、Oracle Grid Infrastructure 11.2.0.2ではこの構成を使用する必要があります。この構成のサポートは、Oracle Grid Infrastructure 11.2.0.3.6以上およびOracle Grid Infrastructure 12cで再開されています。

GDSを使用しないMAA環境でのSCANの使用

グローバル・データ・サービス (GDS) は、インテリジェントなロードバランシングとクライアント・フェイルオーバーの概念をグローバルな分散環境に拡張するOracle Database 12cの新機能ですが、このGDSを使用しないでMaximum Availability Architecture (MAA) 環境を実装して、その環境で (プライマリとスタンバイ両方のサイトの) プライマリ・データベースとスタンバイ・データベースの両方にOracle RACを使用し、Oracle Data Guardを使用してデータベースを同期している場合は、SCANを使用すると、現在アクティブな (プライマリ) データベースがプライマリであるかスタンバイであるかに関係なく、データベース接続に使用できるTNSNAMESの構成を簡素化できます。

この簡素化された構成を使用するために、Oracle Database 11g Release 2では、個々のクライアント接続文字列に使用できる2つの新しいSQL*NETパラメータが導入されています。最初のパラメータは**CONNECT_TIMEOUT**です。このパラメータには、クライアントがOracleデータベースに対してOracle Net接続を確立するときのタイムアウト期間 (秒) を指定します。このパラメータは、SQLNET.ORA内に指定されているSQLNET.OUTBOUND_CONNECT_TIMEOUTよりも優先されます。もう1つのパラメータは**RETRY_COUNT**です。RETRY_COUNTに指定した回数だけADDRESS_LISTのアドレスがチェックされると、接続の試行が終了します。

この2つのパラメータを使用すると、プライマリ・サイトとスタンバイ・サイトの両方のSCANをクライアント接続文字列で使用できます。タイムアウトを指定すると、ランダムに選択されたアドレスが現在アクティブではないサイトのアドレスである場合でも、常軌を逸するほどの長時間が経過するまで待機せずに、接続リクエストをフェイルオーバーさせることができます (オペレーティング・システムによっては、デフォルトのタイムアウト期間が10分にもなります)。

```
sales.example.com =(DESCRIPTION= (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
(ADDRESS_LIST= (LOAD_BALANCE=on)(FAILOVER=ON)
(ADDRESS=(PROTOCOL=tcp)(HOST=sales1-scan)(PORT=1521))
(ADDRESS=(PROTOCOL=tcp)(HOST=sales2-scan)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME= saleservice.example.com)))
```

例12 : MAA環境のTNSNAMES.ORAエントリ

GDSを使用するMAA環境でのSCANの使用

グローバル・データ・サービス (GDS) はOracle Database 12cの新機能で、インテリジェントなロードバランシングとクライアント・フェイルオーバーの概念をグローバルな分散環境に拡張し、複数のフェイルオーバー・ターゲットを使用して可用性を維持できるようにします。GDSを使用するMAA構成では、SCANリスナーは使用されません。GDSとSCANの構成の詳細については、Oracleのそれぞれのドキュメントを参照してください。

Oracle Connection ManagerによるSCANの使用

Oracle RAC DatabaseでOracle Connection Manager（Oracle CMAN）を使用する場合は、Oracle RAC インスタンスのREMOTE_LISTENERパラメータにOracle CMANサーバーを含め、ロードバランシングに関連する情報をOracle CMANサーバーで取得して、使用可能なすべてのインスタンスを対象に接続のロードバランシングが実行されるようにする必要があります。

もっとも簡単にこれを実現するには、例13に示すように、クライアントがOracle CMAN経由で接続するデータベースのREMOTE_LISTENERに、Oracle CMANサーバーをエントリとして追加する方法があります。その場合は、クライアントのTNSNAMES接続記述子からSCANを削除し、Oracle CMANサーバーに必要な構成を追加することも必要になる点に注意してください。詳細については、Oracle CMANのドキュメントを参照してください。

SQL> show parameters listener		
NAME	TYPE	VALUE
listener_networks	string	
local_listener	string	(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=148.87.58.109)(PORT=1521))))
remote_listener	string	stscan3.oracle.com:1521,(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=CMANserver)(PORT=1521))))

例13：Oracle CMANを使用する場合のサーバー側のTNSNAMES.oraの例

まとめと結論

Single Client Access Name (SCAN) は、Oracle Real Application Clusters環境で使用される機能です。SCANを使用すると、クラスタ内で実行中のOracle Databaseにアクセスする際に、クライアントは単一の名前を使用できます。SCANは、最初にOracle Grid Infrastructure製品スイートの一部としてOracle Real Application Clusters（Oracle RAC）11g Release 2で導入され、Oracle RAC 12cにおいて機能が拡張されました。

どのバージョンでも、クラスタにアクセスしてこのクラスタ内のデータベースに接続するために使用する名前が1つであるため、クライアントはEZConnectとシンプルなJDBC Thin URLを使用して、クラスタ内で稼働中のデータベースにアクセスすることが可能です。クラスタ内で動作しているデータベースまたはサーバーの数を意識する必要はなく、リクエストされたデータベースが実際にクラスタ内のどのサーバーで動作していても関係ありません。

クラスタ内を浮動するSCANリスナーにデータベース・インスタンスを登録するためのより安全な方法に加えて、クラスタ内の複数のサブネットのサポートとIPv6のサポートにより、最新バージョンのSCANは、DBaaSまたはプライベート・データベース・クラウドの導入をより柔軟かつ効果的に支援します。



Oracle Single Client Access Name (SCAN)

2013年6月

著者：Markus Michalewicz

共著者：Barb Lundhild

Oracle Corporation

World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065

U.S.A.

海外からのお問い合わせ窓口：

電話：+1.650.506.7000

ファクシミリ：+1.650.506.7200

oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

本文書は情報提供のみを目的として提供されており、ここに記載される内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による黙示的保証を含め、商品性ないし特定目的適合性に関する黙示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクル社は本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

OracleおよびJavaはOracleおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

IntelおよびIntel XeonはIntel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。UNIXは、The Open Groupの登録商標です。0113

Hardware and Software, Engineered to Work Together