DNS設定駆け込み寺

勘所をしっかり押さえる

第4回

馬場達也

DNSは、インターネットでアプリケーションを使用する場合になくてはならない重要なシステムである。しかし、DNSの運用は難しく、管理者もなかなか苦労しているところがあると思う。ここでは、実際にネームサーバ(DNSサーバ)を運用する際に生じるさまざまな疑問に答え、管理者がスムーズにDNSを運用するためのテクニックを紹介する。

Q11 大量の類似したリソースレコードを自動的に生成したい

DNSの逆引きができないことによる接続遅延を防ぐため、DHCPで割り当てるアドレスをすべてDNSに登録しておきたい。しかし、手作業で登録するのは非常に面倒なので、自動的に生成したい。

A BINDの$GENERATE制御ステートメントを使用する

BINDには、複数の類似したリソースレコードを自動的に生成するための「$GENERATE制御ステートメント」が用意されている。$GENERATE制御ステートメントを使用すると、ホスト名やIPアドレスに含まれる一部の数字だけが異なるような大量のリソースレコードを簡単に記述することができる。

例えば、ある範囲のIPアドレスをDHCPで自動的に割り当てている場合には、そのDHCPで割り当てられたアドレスがDNSで逆引きできないと、DNSのタイムアウト待ちで接続が遅れたり、最悪の場合には接続を拒否されたりすることがある。このため、DHCPで割り当てるIPアドレスについても、あらかじめ何らかのホスト名でDNSに逆引き登録しておく必要がある。しかし、大量の

```
$TTL 86400
@       IN      SOA     ns1.example.com.  hostmaster.example.com. (
                                2003101800 ; シリアル番号
                                28800      ; リフレッシュ間隔（秒）
                                7200       ; リトライ間隔（秒）
                                604800     ; ゾーンの有効期間（秒）
                                3600       ; ネガティブキャッシュの有効期間（秒）
                                )
        IN      NS      ns1.example.com.   ; このゾーンのプライマリマスタ
        IN      NS      ns2.example.com.   ; このゾーンのセカンダリマスタ
        IN      MX      10 mx1.example.com.
        IN      MX      20 mx2.example.com.
        IN      A       163.135.0.30
ns1     IN      A       163.135.0.10
ns2     IN      A       163.135.0.11
mx1     IN      A       163.135.0.20
mx2     IN      A       163.135.0.21
www     IN      CNAME   example.com.
$GENERATE 128-254 dhcp-$ A 192.168.0.$
```

リスト1● 正引き用ゾーンデータファイルの記述例（example.comゾーン）

レコードを手作業で登録するには非常に手間がかかる。このような場合に、$GENERATE制御ステートメントを使用すると簡単に記述することができる。

$GENERATE制御ステートメントは、ゾーンデータファイル内で使用することができ、その書式は次のようになっている。

$GENERATE range lhs type rhs

「range」には、リソースレコードに挿入したい数字の範囲が入る。例えば、数字の範囲を「64-128」のように記述すると、ホスト名やIPアドレスの指定した場所に、64から128までの数字を順次挿入して複数のリソースレコードを自動生成する。この際に、数字を2つ飛びなどで挿入したい場合は「64-128/2」のように記述する。また、「lhs」には、リソースレコードのowner名（Aレコードの場合はドメイン名、PTRレコードの場合は逆引き形式のIPアドレス）を記述する。数字を挿入する部分は「$」として記述しておく。そして、「type」には、リソースレコードタイプ（PTR、CNAME、DNAME、A、AAAA、NSのいずれか）を記述する。「rhs」には、リソースレコードのデータ（Aレコードの場合はIPアドレス、PTRレコードの場合はドメイン名）を記述する。ここで、数字を挿入する部分は「$」として記述しておく。TTLおよびネットワーククラス（INなど）は、$GENERATE制御ステートメントでは指定できないので注意しよう。

例えば、「192.168.0.128」～「192.168.0.254」の間のアドレスに対して、「dhcp-128.example.com」～「dhcp-254.example.com」のようなホスト名をDNSに登録しておく場合には、リスト1およびリスト2の最終行のように記述する。このように記述しておけば、登録時の作業が容易になるばかりでなく、後にDHCPで割り当てるIPアドレスの範囲を変更した場合などでも、容易に修正することが可能となる。

```
$TTL 86400
@       IN      SOA     ns1.example.com. hostmaster.example.com. (
                                2003101800 ; シリアル番号
                                28800      ; リフレッシュ間隔（秒）
                                7200       ; リトライ間隔（秒）
                                604800     ; ゾーンの有効期間（秒）
                                3600       ; ネガティブキャッシュの有効期間（秒）
                                )
        IN      NS      ns1.example.com.   ; このゾーンのプライマリマスタ
        IN      NS      ns2.example.com.   ; このゾーンのセカンダリマスタ
10      IN      PTR     ns1.example.com.
11      IN      PTR     ns2.example.com.
20      IN      PTR     mx1.example.com.
21      IN      PTR     mx2.example.com.
30      IN      PTR     example.com.
$GENERATE 128-254 $ PTR dhcp-$.example.com.
```

リスト2● 逆引き用ゾーンデータファイルの記述例（0.135.163.in-addr.arpaゾーン）

Q12 ネームサーバへの問い合わせ頻度やその内容を知りたい

DNSの利用状況を把握するために、ネームサーバへの問い合わせがどのくらいの頻度であるのかを知りたい。また、どのレコードに対する問い合わせが多いのかを知りたい。

A BINDのrndcコマンドを使用して、統計情報や問い合わせ内容をログに出力する

BIND 9では、BIND 9.1.0から問い合わせの統計情報をログに出力する機能を備えている。この統計情報は、リスト3のようにrndcコマンドを発行することにより、named.confファイルのdirectoryサブステートメントで記述したディレクトリの「named.stats」ファイルに出力

することができる。

出力される問い合わせ件数はネームサーバ起動時からの累計なので、1時間に何件の問い合わせを受けているのかを知りたい場合には、1時間ごとに統計情報をダンプし、その差分を計算すればよい。1時間ごとに統計情報をnamed.statsファイルにダンプするには、リスト4のようにcrontabを編集すればよい。

すると、named.statsファイルには、リスト5のように1時間ごとに統計情報がダンプされるようになる。

また、ネームサーバが受けた問い合わせの内容を知りたい場合には、問い合わせロギング機能を有効にすることで、問い合わせの内容をログに出力することができる。この機能を有効にするためには、次のようなrndcコマンドを発行すればよい。この機能を有効にした後で再び無効にするためには、このコマンドを再度発行すればよい。

```
# /usr/sbin/rndc querylog
```

実際に、問い合わせロギング機能が有効となっているかどうかを調べるためには、リスト6のようにrndcコマ

```
# /usr/sbin/rndc stats ―― 統計情報を「named.stats」ファイルにダンプ
# cat /var/named/named.stats
+++ Statistics Dump +++ (1062907200)
success 132 ―― 回答を返却した問い合わせの数
referral 0 ―― 回答を返却せず、参照先のみを返却した問い合わせの数
nxrrset 1628 ―― ドメイン名は存在したが、レコードタイプが存在しなかった問い合わせの数
nxdomain 48 ―― ドメイン名が存在しなかった問い合わせの数
recursion 1797 ―― 反復問い合わせを必要とした問い合わせの数
failure 4 ―― 上記以外のエラーを返却した問い合わせの数
--- Statistics Dump --- (1062907200)
```

リスト3● rndcコマンドを使用して統計情報を「named.stats」ファイルにダンプする

```
# crontab -e ―― crontabを編集
0 * * * * /usr/sbin/rndc stats ―― この行を追加
```

リスト4● crontabを編集して1時間ごとに統計情報をダンプする

```
# cat /var/named/named.stats
+++ Statistics Dump +++ (1062907200)
success 132
referral 0
nxrrset 1628
nxdomain 48
recursion 1797
failure 4
--- Statistics Dump --- (1062907200)
+++ Statistics Dump +++ (1062910800)
success 150
referral 0
nxrrset 1748
nxdomain 48
recursion 1929
failure 4
--- Statistics Dump --- (1062910800)
```

リスト5● 1時間ごとにダンプした場合の「named.stats」ファイルの内容

ンドを発行することで確認できる。ここで「query logging is ON」と表示されていれば、ネームサーバが受けた問い合わせの内容がsyslog経由で出力されていることになる。ただし、ネームサーバを再起動すると、この問い合わせロギング機能は自動的にOFFになるので注意しよう。

それでは、実際にログの出力を見てみよう（リスト7）。Red Hat Linux 9の場合は、問い合わせ内容が「/var/log/messages」に出力される。

また、この問い合わせログを指定したファイルに出力したい場合は、named.confファイルにリスト8のようなloggingステートメントを追加すればよい。

これにより、named.confファイルのdirectoryサブステートメントで記述したディレクトリに、loggingステートメントで指定したファイル（リスト8の例では「queries.log」ファイル）が作成され、問い合わせ内容が記録されるようになる。この設定をした場合には、ネームサーバ起動後に問い合わせロギング機能が自動的に有効となる。

NTTデータ　馬場達也

```
# /usr/sbin/rndc status
number of zones: 6
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is ON ―――――――――――――――――――――――――――― 問い合わせロギングが有効となっている
server is up and running
```

リスト6●　問い合わせロギングが有効になっていることを確認

```
# cat /var/log/messages | grep query
Sep  7 11:26:53 ns1 named[2052]: client 192.168.0.64#1078: query: www.idg.co.jp IN A
Sep  7 11:27:00 ns1 named[2052]: client 192.168.0.64#1081: query: adnet.asahi.com IN A
Sep  7 11:27:05 ns1 named[2052]: client 192.168.0.64#1090: query: ads.asah.valueclick.jp IN A
Sep  7 11:39:11 ns1 named[2052]: client 127.0.0.1#33030:   query: 64.0.168.192.in-addr.arpa IN PTR
Sep  7 11:41:00 ns1 named[2052]: client 127.0.0.1#33030:   query: 64.0.168.192.in-addr.arpa IN PTR
Sep  7 11:44:54 ns1 named[2052]: client 192.168.0.64#1180: query: www.nttdata.co.jp IN A
Sep  7 11:44:58 ns1 named[2052]: client 127.0.0.1#33030:   query: 64.0.168.192.in-addr.arpa IN PTR
```

リスト7●　問い合わせログの出力例

```
logging {
        channel query_log {
                file "queries.log"; ―――――――――――――――――― ログを出力するファイル名を指定
                print-time yes; ―――――――――――――――――――― ログに時刻を出力するように指定
        };
        category queries { "query_log"; }; ―――――――――――― 問い合わせ内容をログに出力するように指定
};
```

リスト8●　「named.conf」ファイルで追加するloggingステートメントの例