DNS設定駆け込み寺 第6回

勘所をしっかり押さえる

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 ネームサーバを再起動させずにゾーンデータを更新したい

ゾーンデータの更新内容を有効にするためには、ネームサーバを再起動すればよい。しかし、ネームサーバを再起動すると、キャッシュデータが消えてしまうため、ネームサーバを再起動せずに必要なゾーンデータのみを再度読み込ませたい。

A 「rndc reconfig」コマンドまたは「rndc reload」コマンドを使用する

ゾーンデータを更新したり、新しいゾーンを追加したりした場合にネームサーバを再起動すると、キャッシュの内容が消えてしまう。また、ゾーンの数が多い場合には、ネームサーバが起動するまでに時間がかかることがある。rndcコマンドを使用すると、ネームサーバを再起動することなく目的のゾーンデータのみを更新できる。

新しいゾーンを追加したり削除したりした場合は、ネームサーバが起動している状態で、次のように「rndc reconfig」コマンドを発行すればよい。

```
# /usr/sbin/rndc reconfig
```

このコマンドを発行することにより、更新されたnamed.confファイルの内容が読み込まれ、新たに追加されたゾーンのゾーンデータファイルの内容が読み込まれる。このコマンドでは、既存のゾーンデータファイルの内容は更新されない。

また、あるゾーンデータファイルのみを更新した場合は、次のように、再度読み込ませたいゾーンを指定して「rndc reload」コマンドを発行すればよい。

```
# /usr/sbin/rndc reload example.com
```

ただし、BINDのビュー機能を使用している場合は、次のようにクラスとビュー名を指定して「rndc reload」コマンドを発行する必要がある。

```
# /usr/sbin/rndc reload example.com in internal
                                    クラス  ビュー名
```

新たに追加したゾーンを含むすべてのゾーンデータファイルをリロードしたい場合は、次のように、ゾーン名を指定せずに「rndc reload」コマンドを発行すればよい。このコマンドを発行することにより、named.confファイルで定義されているすべてのゾーンが再度読み込まれる。

```
# /usr/sbin/rndc reload
```

また、ゾーンの更新が行われた場合には、セカンダリネームサーバには、プライマリネームサーバからNOTIFYメッセージが送信されると同時にゾーン転送が行われる。このため、通常は、セカンダリネームサーバでは更新に関する作業を行う必要はない。しかし、NOTIFYメッセージが届かないなどの理由でゾーン転送が行われない場合もある。このような場合には、セカンダリネームサーバ上で、次のようにゾーン転送を開始したいゾーン名を指定して「rndc refresh」コマンドを発行すればよい。これにより、次のゾーン転送が開始されるのを待たずに、すぐにゾーン転送を行うことができる。ただし、

プライマリネームサーバで更新したゾーンのシリアル番号がアップしていないとゾーン転送は開始されないので、ゾーンデータファイルを更新した場合には、シリアル番号の更新も忘れないようにしよう。

```
# /usr/sbin/rndc refresh example.com
```

Q12 ネームサーバを再起動させずにキャッシュを消去したい

まちがって設定したレコードがすでにキャッシュされてしまった場合に、ネームサーバを再起動させずにそのキャッシュデータを消去したい。

A 「rndc flush」コマンドを使用する

キャッシュされているレコードを消去したい場合には、ネームサーバを再起動すればよい。しかし、rndcコマンドを使用すれば、ネームサーバの再起動なしでキャッシュの内容を消去することができる。

まず、キャッシュされているデータを確認してみよう。キャッシュされているデータを見るには、リスト1のように「rndc dumpdb」コマンドを発行すればよい。このコマンドを発行すると、デフォルトでは、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_dump.db」ファイルにキャッシュの内容がダンプされる。ダンプファイルのファイル名は、named.confファイルのoptionsステートメントにおいて、「dump-file」サブステートメントを記述することにより変更することもできる。

このキャッシュの内容を消去するためには、次のように「rndc flush」コマンドを発行すればよい。ただし、現在のところ、キャッシュの内容をレコードごとに消去することはできない。このコマンドを発行するとすべてのキャッシュデータが消されるので注意しよう。

```
# /usr/sbin/rndc flush
```

また、特定のビューにおけるキャッシュを消去する場合には、次のようにビュー名を指定して「rndc flush」コマンドを発行すればよい。

```
# /usr/sbin/rndc flush internal
                      ↑
                      └ビュー名
```

```
# /usr/sbin/rndc dumpdb
# cat /var/named/named_dump.db
;
; Cache dump of view '_default'
;
$DATE 20031109073857
; authanswer
.                       518361  IN NS   A.ROOT-SERVERS.NET.
                        518361  IN NS   B.ROOT-SERVERS.NET.
                        518361  IN NS   C.ROOT-SERVERS.NET.
                        518361  IN NS   D.ROOT-SERVERS.NET.
                        518361  IN NS   E.ROOT-SERVERS.NET.
                        518361  IN NS   F.ROOT-SERVERS.NET.
                        518361  IN NS   G.ROOT-SERVERS.NET.
                        518361  IN NS   H.ROOT-SERVERS.NET.
                        518361  IN NS   I.ROOT-SERVERS.NET.
                        518361  IN NS   J.ROOT-SERVERS.NET.
                        518361  IN NS   K.ROOT-SERVERS.NET.
                        518361  IN NS   L.ROOT-SERVERS.NET.
                        518361  IN NS   M.ROOT-SERVERS.NET.
```

リスト1● キャッシュデータのダンプの例。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_dump.db」ファイルにキャッシュの内容がダンプされる

```
; authauthority
                        10761   ¥-A     ;-$
; glue
jp.                     172782  NS      A.DNS.jp.
                        172782  NS      B.DNS.jp.
                        172782  NS      C.DNS.jp.
                        172782  NS      D.DNS.jp.
                        172782  NS      E.DNS.jp.
                        172782  NS      F.DNS.jp.
; authauthority
idg.co.jp.              583     NS      dns11.cwidc.net.
                        583     NS      dns22.cwidc.net.
; authanswer
www.idg.co.jp.          583     A       210.134.87.3
; glue
A.DNS.jp.               172782  A       61.120.151.100
; glue
B.DNS.jp.               172782  A       202.12.30.131
; glue
C.DNS.jp.               172782  A       165.76.0.98
; glue
D.DNS.jp.               172782  A       202.232.2.34
; glue
E.DNS.jp.               172782  A       192.50.43.53
; glue
F.DNS.jp.               172782  A       150.100.2.3
～（以下省略）～
```

リスト1の続き

Q13 キャッシュのサイズを制限したい

ネームサーバのメモリが少ないので、キャッシュに使用するメモリの量を制限したい。

A optionsステートメントでキャッシュの設定を行う

BINDのデフォルトでは、キャッシュのサイズは制限されておらず、期限切れになった（リソースレコードのTTLが0になった）キャッシュデータの削除は60分ごとに行われる。BIND 9.2.0以降であれば、キャッシュのサイズの上限値を、named.confファイルのoptionsステートメントの「max-cache-size」サブステートメントで設定できる。これを設定することで、キャッシュが上限サイズに達した場合は古いデータをキャッシュから削除し、キャッシュに使用するメモリ量を圧迫しない。

max-cache-sizeサブステートメントではキャッシュサイズの上限値をバイト単位で指定することができるが、設定できる値は2MB以上という制限がある。また、「cleaning-interval」サブステートメントを使用することにより、期限切れになったキャッシュデータの削除間隔を設定できる。デフォルトでは、期限切れになったキャッシュデータであっても、60分ごとにしかメモリからそのデータが削除されることはない。しかし、この間隔を短くすることで、期限切れになったキャッシュデータをすぐに削除できるようになるため、メモリを効率よく利用できる。

例えば、キャッシュサイズの上限値を10MB、期限切れとなったレコードをチェックする間隔を15分に変更する場合は、次のように設定する。

```
options {
        directory "/var/named/";
        max-cache-size 10m;
```

キャッシュサイズの上限値を10MBに設定

```
    cleaning-interval 15;
                ↑期限切れレコードのチェック間隔を15分に設定
};
```

また、キャッシュサイズを小さくする方法として、「max-cache-ttl」サブステートメントと「max-ncache-ttl」サブステートメントを使用する方法もある。max-cache-ttlサブステートメントは、キャッシュされたリソースレコードの最大保持時間（秒）を指定するためのものであり、デフォルトでは1週間（60万4,800秒）となっている。max-ncache-ttlサブステートメントは、ネガティブキャッシュの最大保持時間（秒）を指定するためのもので、デフォルトでは3時間（1万800秒）となっている。これらの値を短く設定することにより、取得したリソースレコードのTTLが設定値より大きい場合でも、そのネームサーバ上では設定した時間しかキャッシュとして保持しなくなるため、キャッシュのサイズを小さく保つことができる。しかし、max-cache-ttlサブステートメントで設定する値は1以上にする必要がある。もし0に設定すると、クライアントには名前解決の結果が返らなくなり、代わりに「SERVFAIL」エラーが返ることになるので注意しよう。

例えば、キャッシュされたリソースレコードの最大保持時間を8万6,400秒（1日）、ネガティブキャッシュの最大保持時間を3,600秒（1時間）に変更する場合は、次のように設定する。

```
options {
    directory "/var/named/";
    cleaning-interval 15;
        ↑期限切れレコードのチェック間隔を15分に設定
    max-cache-ttl 86400;
        ↑キャッシュされたレコードの最大TTL値を1日に設定
    max-ncache-ttl 3600;
        ↑ネガティブキャッシュの最大TTL値を1時間に設定
};
```

アクセスして得られたレコードのTTL値やネガティブキャッシュのTTL値が、max-cache-ttlサブステートメントやmax-ncache-ttlサブステートメントで設定した最大保持時間を超えていた場合には、そのTTL値はmax-cache-ttlサブステートメントやmax-ncache-ttlサブステートメントで設定した値に置き換えられてクライアントに返される。例えば、存在しないレコードを問い合わせた場合は、リスト2のようにSOAレコードが返答されるが、このSOAレコードではネガティブキャッシュのTTL値が8万6,400秒となっているにもかかわらず、SOAレコードのTTL値がmax-ncache-ttlサブステートメントで設定した3,600秒となってクライアントに返される。

NTTデータ　馬場達也

```
$ dig www.example.dom

; <<>> DiG 9.2.1 <<>> @192.168.0.1 www.example.dom
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 64001
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;www.example.dom.               IN      A

;; AUTHORITY SECTION:
.                       3600    IN      SOA     A.ROOT-SERVERS.NET. NSTLD.VERISIGN-GRS.COM. 2003110900 1800 900 604800 86400
;; Query time: 197 msec                                  SOAレコードのネガティブキャッシュTTLは86400
;; SERVER: 192.168.0.1#53(192.168.0.1)                   となっているが、実際のTTLは3600となる
;; WHEN: Mon Nov 10 00:26:40 2003
;; MSG SIZE  rcvd: 108
```

リスト2●　max-ncache-ttlサブステートメントでネガティブキャッシュの最大TTL値を3,600秒に設定した場合のdigの結果。SOAレコードのTTL値が3,600秒になっている