NSEC/NSEC3のType Bit Mapsについて

DNS BoF 2019
2019/11/28
Toshifumi Sakaguchi

自己紹介

- 坂口　俊文
- Twitter: @siskrn
- GitHub: https://github.com/sischkg/
- DNS Summer Day, DNSOPS.JP BoFで発表 https://dnsops.jp/

Agenda

1. NSEC/NSEC3のType Bit Mapsの説明
2. PowerDNS RecursorでのType Bit Mapsの実装の問題
3. Type Bit Mapsのテキスト表現
4. Type Bit Mapsとゾーン転送・キャッシュのダンプの話
5. まとめ

Type Bit Maps

NSECレコードは以下の項目を示します。

1. OwnerとNextの間にドメイン名は存在しないこと
2. OwnerにはType Bit Mapsに書かれたタイプのみ存在すること

```
dnsops.jp.   10800  IN  NSEC  _443._tcp.dnsops.jp. A NS SOA MX AAAA RRSIG NSEC DNSKEY
CAA
```

- dnsops.jp.と_443._tcp.dnsops.jp.の間にはドメイン名は存在しない
- dnsops.jp.には、A NS SOA MX AAAA RRSIG NSEC DNSKEY CAAレコードが存在する

Type Bit MapsのWire Format(1)

Typeの上位８bitsをWindow Blockで、下位8bitsをBit Mapで表現

- 0-255のType:
  - Window Block = 0
  - 存在するTypeに対応するBit MapのBitを１
- 256-511のType:
  - Window Block = 1
  - 存在するTypeに対応するBit MapのBitを１
- …

Window Block								Bit Map Length								Bit Map															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Type Bit MapsのWire Format(2)

Types = A(1), AAAA(28), CAA(257)の場合

（本来はRRSIGなども含まれるが紙面の都合で省略）

0x00	0x04	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0
Window Block	bit map length	0	**1**	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	**28**	29	30	31

0x01	0x01	0	**1**	0	0	0	0	0	0
Window Block	bit map length	256	**257**	258	259	260	261	262	263

Type Bit MapsのWire Format

Type Bit MapsのWire Formatは、サイズが小さくなるように定義。

PowerDNS Recursorのメモリ使用量の問題

発端

- PowerDNS Recusorを例のファジングにかけると数日後にOOMで強制終了
- PowerDNS Recusorの特徴
  - キャッシュのエントリ数を制限する機能(max-cache-entries)→あり
  - キャッシュのメモリ容量を制限する機能→なし
- メモリ使用量の大きいレコードをキャッシュすると、エントリ数の制限以下でメモリを使い切る

PowerDNS Recursorのメモリ使用量の問題

メモリ使用量の大きいレコードは？

- ヘッダファイル “dnsrecords.hh” からメモリ使用量が大きいclassを探す
- NSECおよびNSEC3

```
class NSECRecordContent : public DNSRecordContent
{
// snip
  DNSName d_next;
  std::set<uint16_t> d_set;
private:
};
```

https://github.com/PowerDNS/pdns/blob/rec-4.1.14/pdns/dnsrecords.hh#L506

PowerDNS RecursorにおけるType Bit Mapsの実装

- PowerDNS RecusorはType Bit MapsをC++のstd::set<uint16_t>へ保存
- Type Bit Mapsの1bitとstd::setの1エントリが対応
- つまり全bit(65336bits) を 1⇔ std::setに 65536エントリ
- std::setはRed-Black treeを用いて実装(CentOS 7.6のGCC 4.8.5)

メモリ使用量の予測

Red-Black treeのノード(CentOS 7.7 x86_64)は40bytes

sizeof(_Rb_tree_node<uint16_t>) == 40

Type(2bytes)のデータを保持するために、40bytes消費

```
struct _Rb_tree_node_base {
     _Rb_tree_color _M_color;              // color                4 bytes
     _Base_ptr _M_parent;                  // pointer of parent node     8 bytes
     _Base_ptr _M_left;                    // pointer of left node          8 bytes
     _Base_ptr _M_right;                   // pointer of right node         8 bytes
}
// snip
struct _Rb_tree_node : public _Rb_tree_node_base
{
      typedef _Rb_tree_node<_Val>* _Link_type;
      _Val _M_value_field;                    // Type                2 bytes
```

Type Bit Mapsのサイズ

全てのTypeが存在するときのType Bit Mapsのサイズ・メモリ使用量

DNS Message内のサイズ

= <bit map count> x (<Window Block> + <Bit Map Length> + <Bit Map>)

= 256 x (1 + 1 + 32) bytes

= **8704 bytes**

PowerDNS Recursorでのメモリ使用量

= <node size of red-black tree> x 65536 + <Overhead> bytes

= 40 x 65536 + <Overhead> bytes

= 2,621,400 + <Overhead> bytes

~ 3MB（実測値）

サンプルコード(https://github.com/sischkg/huge_nsec_response/blob/master/set-uint16_t-x100.cpp)

PowerDNS Recursorのメモリ使用量の問題

キャッシュの制限機能

- BINDやUnboundではキャッシュの量をメモリ使用量で制限
- PowerDNS Recursorではエントリ数で制限

PowerDNS Recursorのキャッシュを細工したNSECでレコードで満たすと、
非常に多くのメモリを使用

デフォルトの制限値: max-cache-entries=1,000,000

推定メモリ使用量: 3MB x 1,000,000 ～ 3TB

PowerDNS Recursorのメモリ使用量の問題

キャッシュの制限値を3MB/エントリをもとに設定すると、

十分なキャッシュを保持することができない

キャッシュサーバのメモリ: 512GB

キャッシュ可能なエントリ数: 512GB / 3MB = 170,666

PowerDNS Recursorのメモリ使用量の問題

対策:　PowerDNS Recusor 4.2.0にバージョンアップ

- 4.2.0ではType Bit MapsのType数が200に達すると、コンテナをstd::set<uint16_t>からstd::bitset<65536>へ変更
- Pull Request:Use a bitset for NSEC(3) records with lots of types #7345 (https://github.com/PowerDNS/pdns/pull/7345/commits/27d4a65bf8d908b02ea84e662683d31b077306ab)
- std::bitsetを利用することで、メモリ使用量をエントリあたり2KB程度まで減少

PowerDNS Recursorのメモリ使用量の問題のまとめ

- PowerDNS RecursorはType Bit Mapsをキャッシュするときのメモリ使用量が大きい
- エントリ数制限値が大きい場合は、メモリを使い切る可能性がある
- 4.2.0で修正

Type Bit Mapsのテキスト表現

テキスト表現

- dig/drillなどの出力
- ゾーンファイル
- キャッシュのダンプ

NSECのテキスト表現の例

```
example.com. IN NSEC  dns01.example.com. A NS SOA MX
RRSIG NSEC DNSKEY
```

Type Bit Mapsのテキスト表現

Type Bit Mapsの全てのbitを1としたNSECレコード

```
example.com. 3600 IN NSEC a.example.com. RESERVED0 A NS MD MF CNAME SOA MB MG MR NULL WKS PTR
HINFO MINFO MX TXT RP AFSDB X25 ISDN RT NSAP NSAP-PTR SIG KEY PX GPOS AAAA LOC NXT EID NIMLOC
SRV ATMA NAPTR KX CERT A6 DNAME SINK OPT APL DS SSHFP IPSECKEY RRSIG NSEC DNSKEY DHCID NSEC3
NSEC3PARAM TLSA SMIMEA TYPE54 HIP NINFO RKEY TALINK CDS CDNSKEY OPENPGPKEY CSYNC
...
TYPE65530 TYPE65531 TYPE65532 TYPE65533 TYPE65534 TYPE65535
```

https://raw.githubusercontent.com/sischkg/huge_nsec_response/master/nsec_response.txt

1レコード当たり約 640KB

ゾーン転送で受信したゾーン

BINDでは、ゾーン転送で受信したゾーンのファイル形式

- 9.8.0未満: テキスト
- 9.8.0以上で”masterfile-format text;”: テキスト
- 9.8.0以上で”masterfile-format”未指定: raw

BIND9の最新動向(https://dnsops.jp/event/20120901/BIND9_update-1.pdf)

1000個のNSECレコードを持つゾーンファイルのサイズ

- テキスト形式:　645 MB
- raw形式:　　8.8 MB

キャッシュのダンプ

フルリゾルバでは、以下のようにキャッシュの内容を出力することが可能

BIND:

```
rndc dumpdb
```

Unbound:

```
unbound-control dump_cache > /tmp/dumpdb.txt
```

PowerDNS Recursor:

```
rec_control dump-cache /tmp/dumpdb.txt
```

キャッシュのダンプ

1000レコード当たりのダンプファイルのサイズ

- BIND: 約640MB
- Unbound: 約1.1MB
- PowerDNS Recursor: 約1.3GB

想定したサイズより大きいダンプファイルが生成される可能性

- BINDのAdiministrator Referece Manualに記載を依頼
- rndc dumpdb output may be larger than expected - documentation update request(https://gitlab.isc.org/isc-projects/bind9/issues/795)

キャッシュのダンプ(Unbound)

Unboundでは、Type Bit Mapsをすべて出力しないため、ダンプファイルが小さい

```
1.example.com.  3215    IN      NSEC    a.1.example.com. A NS MD
MF CNAME SOA MB .... TYPE152 TYPE1;rrset 10415 1 1 7 0
```

キャッシュのダンプ(PowerDNS Recursor)

PowerDNS Recosor:

- キャッシュをダンプ中に応答しない時がある
- Threadごとにキャッシュを持っているらしく、同じレコードが複数回出力されるため、ダンプファイルが大きい

rec_control(https://doc.powerdns.com/recursor/manpages/rec_control.1.html)

Dumps the entire cache to *FILENAME*. This file should not exist already, PowerDNS will refuse to overwrite it. **While dumping, the recursor will not answer questions.**

Typical PowerDNS Recursors **run multiple threads, therefore you'll see duplicate**, different entries for the same domains. The negative cache is also dumped to the same file. The per-thread positive and negative cache dumps are separated with an appropriate comment.

Type Bit Mapsのテキスト表現のまとめ

- Type Bit Mapsのテキスト表現は、非常に大きなサイズになる場合がある
- ゾーン転送後のゾーンファイルや、キャッシュのダンプファイルのサイズも大きくなる
- ダンプファイルやゾーンファイルの保存先は、ファイルシステムに十分な余裕が必要

まとめ

- PowerDNS Recursorは特殊なNSEC/NSEC3をキャッシュするときのメモリ使用量が大きい
- エントリ数制限値が大きい場合は、メモリを使い切る可能性がある
- Type Bit Mapsのテキスト表現は、非常に大きなサイズになる場合がある
- そのためゾーン転送後のゾーンファイルや、キャッシュのダンプファイルのサイズも大きくなる
- ただし、特殊なNSEC/NSEC3を作成して実際に実行する人が存在するかは不明