Unbound キャッシュDNS サーバ 大規模用途向け機能の実装

東 大亮

アジェンダ

- Unboundの簡単な紹介
- 大規模用途向け機能の実装
  - DNSラウンドロビン
  - minimal-responses
- まとめ

Unboundとは(1)

- **蘭NLnet Labsが開発・保守を行っているキャッシュDNSサーバ**
  - 本家Web http://unbound.net
  - BSDスタイルライセンスで配布、Unix/Windows(installerも配布)で動作
  - “BIND alternative”を謳い文句に2008年正式リリース、最新版 1.4.18

日本Unboundユーザ会もおすすめ！
http://unbound.jp (日本語ドキュメント多数)

- **キャッシュDNSサーバ専用。一通りの機能は備える**
  - DNSフルリゾルバサービス
    - IPv4/IPv6デュアルスタック、ソースIPアドレスによるアクセス制限
    - TXID/Port Randomization等のキャッシュポイズニング対策
  - DNSSEC validator ー BIND以外のfreeなキャッシュDNSサーバの中では唯一
  - 簡易的な権威DNSサーバ機能、フォワーダ

Unboundとは(2)

- 性能

		キャッシュヒット率	
		0%	100%
Unbound 1.4.18	qps	9,947	54,527
BIND 9.8.3-P2	qps	3,730	20,106
Unbound/BIND	qps/qps	2.67	2.71

CPU: Core2duo T5500 1.66GHz Mem: 2GB OS: CentOS 5.8 (32bit)
キャッシュヒット率100%のケースは、メモリ上に100万エントリをキャッシュさせた状況。キャッシュヒット率0%のケースは、キャッシュが空の状態から、各クエリにつき権威サーバへの問い合わせは１回のみ発生させながらキャッシュを満たしていく状況を作った。

- 脆弱性の発生状況

		2012年	2011年	2010年	2009年	2008年
Unbound	件数	1	2	1	1	0

セキュリティ脆弱性が発表された回数。2012年の１件はGhost Domain Names問題で、当時の最新版ではすでに修正済。

Unboundに欠けている主な機能 (BIND9キャッシュに比べて*)

Unboundに有ってBIND9に無い機能も多数ですが敢えて・・・

* BIND9.8.x vs Unbound 1.4.16

- View
- ResponsePolicyZone**、AAAA Filter、DNS64***
- DNSラウンドロビン
- minimal-responses

** RPZは簡易権威サーバ機能で代替可能な場合が多い

*** DNS64は過去の版向けパッチは存在

Unboundに欠けている主な機能 (BIND9キャッシュに比べて*)

Unboundに有ってBIND9に無い機能も多数ですが敢えて・・・

* BIND9.8.x vs Unbound 1.4.16

- View
- ResponsePolicyZone**、AAAA Filter、DNS64***
- DNSラウンドロビン
- minimal-responses

大規模環境のBIND9をUnboundに置き換える場合に問題となり得る

➡**実装してみた**

** RPZは簡易権威サーバ機能で代替可能な場合が多い

*** DNS64は過去の版向けパッチは存在

DNSラウンドロビン

- DNS問合せ対象のドメインに同一タイプの複数のレコード(RRSet)が付与されていた場合、**応答するRRSetの順番を問合せ毎に変化させる**機能
  - DNS検索で複数のレコードがある場合、**最初のレコードへアクセスする端末が多いため、結果的にサーバ負荷分散の効果**が得られる
- ラウンドロビンはDNSキャッシュとして利用者が多い**BINDでは古くからデフォルト動作**であり、設定も簡単のため、**簡易なサーバ負荷分散テクニックとして広く使われている**

1回目
```
$ dig unbound.net +short
213.248.210.39
213.154.224.1
```

2回目
```
$ dig unbound.net +short
213.154.224.1
213.248.210.39
```

3回目
```
$ dig unbound.net +short
213.248.210.39
213.154.224.1
```

4回目
```
$ dig unbound.net +short
213.154.224.1
213.248.210.39
```

BIND9のキャッシュDNSサーバに問い合わせるたびに、応答のAレコード(IPアドレス)の順番が入れ替わっている

UnboundとDNSラウンドロビン(1)

- Unboundは、長らくDNSラウンドロビンを実装しなかった
  - 権威サーバから得た答えをキャッシュした時のRRSetの順番を保持して応答
  - １つのUnboundサーバを共用している端末群は**DNSラウンドロビンによる負荷分散が効かず、特定のホストに負荷が集中する**

1回目
```
$ dig unbound.net +short
213.248.210.39
213.154.224.1
```

２回目
```
$ dig unbound.net +short
213.248.210.39
213.154.224.1
```

３回目
```
$ dig unbound.net +short
213.248.210.39
213.154.224.1
```

４回目
```
$ dig unbound.net +short
213.248.210.39
213.154.224.1
```

Unboundでは、キャッシュが消えるまで常にRRSet内の順序変化なし

DNSラウンドロビンによる負荷分散を期待する**コンテンツ事業者が困る**のはもちろん、負荷集中でDNSキャッシュサーバを運用する**ISP事業者にとってもユーザ不満足につながる**（多数の端末を収容する**大規模用途で問題**となり得る）

当初からUnboundにDNSラウンドロビンの実装を望む声多数

UnboundとDNSラウンドロビン(2)

- 2012年3月にある人がDNSラウンドロビンのパッチをUnbound-users MLに投稿
  - 軽量で評判も上々だったが、**RRSetの順番を入れ替える乱数源として、スレッドセーフでないrandom()関数を使用**していたため、メインラインに取り込まれず
  - Unbound自身もスレッドセーフな方法で疑似乱数を生成するルーチンを持っているが、DNS応答を生成する（同時にRRSetの順序を入替る）部分のコードからそれにアクセスするためには大改造が必要

UnboundとDNSラウンドロビン(3)

- 私（東）が行った**DNSラウンドロビンパッチの改良**
  - RRSetの順番を入れ替える**乱数源として、DNSメッセージ中のクエリIDを利用**
    - クエリIDはblind attackによるDNS毒入れ攻撃防止のために予測不能な乱数~~というタテマエ~~
  - **疑似乱数生成処理に伴う性能低下やコードの大改造無く、DNSラウンドロビンを実現**
- **あっさりメインラインに取込まれ、Unbound 1.4.17で正式リリース！**

残念ながらデフォルトでオフなので、DNSラウンドロビンするためには以下の設定を unbound.confに入れてください

```
server:
    rrset-roundrobin: yes
```

脱線：RFC3484で、DNSラウンドロビンは廃れる方向？(1)

- RFC3484 Default Address Selection for IPv6
  - クライアントとサーバ間のソースと宛先IPアドレスの組合せに複数の選択肢がある場合に、どのアドレスを使うか決定するルールを規定　(for IPv6と言っているがIPv4にも適用)
    - **DNS検索で複数のサーバIPアドレスが得られた場合は、クライアントIPアドレスの先頭から最も長くbitがマッチするサーバアドレスを優先する**(Section6, Rule 9: Longest match)

例:

クライアントIP: **192.168.0.1**／サーバIP: 192.0.2.1, 192.168.2.1

➡ DNSラウンドロビンに関わらず、

クライアントIPとlongest matchな 192.168.2.1を優先する

脱線：RFC3484で、DNSラウンドロビンは廃れる方向？(2)

- RFC3484を実装するシステムが現れてきたので、**今後DNSラウンドロビンによる負荷分散が使われなくなる**という意見あり
  - RFC3484式宛先アドレス選択を実装: glibc (GNU/Linux), Windows Vista*

*Vistaのgethostbyname (getaddrinfoの前身の古い名前解決関数)では、最小のIPアドレスを優先するという情報あり。
http://support.microsoft.com/kb/948505/en

- **そうでもないかも？**
  - 従来通りDNS検索結果の最初のIPアドレスを利用するシステムも依然多数、現実としてDNSラウンドロビンしてるWebサーバ群で、少々の偏りはあるものの概ねアクセス分散が効いているように見える
  - Windowsは、Vistaで一度RFC3484式のアドレス選択を実装したが、Win7でXP以前の動作(最初のIPアドレスを優先)に戻した経緯あり
  - **RFC3484端末に対しても、DNSラウンドロビンが効くような構成は可能**

脱線：RFC3484で、DNSラウンドロビンは廃れる方向？(3)

- CDN・CSP事業者の中には、**DNS応答するA/AAAA RRSet の上位プリフィックスを揃える**ことで、RFC3484端末からのアクセスでも偏りにくくするテクニックが見られる

$ dig www.youtube.com +short

youtube-ui.l.google.com.
173.194.38.105
173.194.38.110
173.194.38.96
173.194.38.97
173.194.38.98
173.194.38.99
173.194.38.100
173.194.38.101

上位３オクテットが揃っていることに注意。クライアントIPが173.194.38.*の範囲になければ、**これらのサーバアドレスはRFC3484 Section6, Rule9 (longest match)的には全員引分け。**次のRule10により、**クライアントはRRSetの最初のアドレスを利用する(=DNSラウンドロビン有効)**

minimal-responses (1)

DNS応答の構造

- DNS応答メッセージは、**QUESTION**、**ANSWER**、**AUTHORITY**、**ADDITIONAL** の4セクションからなる

- DNS問合せに直接答えるのが**ANSWER**セクション

- **AUTHORITY**、**ADDITIONAL**は、答えを教えてくれた権威サーバ等の付加的情報が入る

BIND9/Unboundのデフォルト動作

```
$ dig dnsops.jp A
(略)
;; QUESTION SECTION:
;dnsops.jp.   IN  A

;; ANSWER SECTION:
dnsops.jp.    IN A   210.171.226.61

;; AUTHORITY SECTION:
dnsops.jp.  IN NS   ns1.dnsops.jp.
dnsops.jp.  IN NS   ns2.dnsops.jp.

;; ADDITIONAL SECTION:
ns1.dnsops.jp.    IN A       210.171.226.61
ns2.dnsops.jp.    IN A       183.181.160.83
;; MSG SIZE  rcvd: 111
```

minimal-responses (2)

- minimal-responsesは、**AUTHORITY、ADDITIONAL**セクションが規格上必須でない場合は**省略***する機能（キャッシュサーバの肯定的応答では概ね必須でない）
- **DNS応答**のメッセージサイズが**小さくなる**
- BIND9ではデフォルトではmin-respはオフだが、オンの実装も多い
  - オンの実装：PowerDNS recursor、Google Public DNS、(djb)dnscache等

“minimal-responses yes”なBIND9/Unboundの応答

```
$ dig dnsops.jp A
(略)
;; QUESTION SECTION:
;dnsops.jp.   IN  A

;; ANSWER SECTION:
dnsops.jp.    IN A    210.171.226.61

;; (AUTHORITY/ADDITIONAL 無し)

;; MSG SIZE  rcvd: 43
```

DNS応答サイズが激減
(111bytes➡43bytes)

*EDNS0応答の場合、ADDITIONALセクションのOPTレコードだけは残します

minimal-responses (3)

多数の端末を収容する大規模なDNSキャッシュサーバの用途では、minimal-responsesがいろいろ有利

- **ネットワーク帯域の削減**
  - DNSでもクエリ数が大きいとネットワーク帯域が無視できなくなる。応答サイズが小さければ、必要帯域も小さくなる。
- **TCPフォールバックの可能性を減らす**
  - （EDNS0なしの）DNS over UDPの応答は512バイトが上限。それを超える場合はTCPでDNSクエリをやりなおす必要があるが、**TCPクエリを（EDNS0も）実装しない端末が多数存在する**
  - つまり、**DNS応答サイズが大きいドメインは、端末によっては引けないことがある**
  - **minimal-responsesで応答サイズが小さくしておけば、TCPが必要になる可能性が減る** (100%問題を防げるわけではありません)
- **DNSキャッシュサーバの応答性能が向上することがある**
  - BIND9では1.5倍程度の向上、Unboundでは数%向上

minimal-responses (4)

- **minimal-responsesがおすすめの用途**
  - PCやスマートフォン等、A/AAAAクエリが多数を占める端末がDNSキャッシュサーバの主要なクライアントの場合
    - authority/additional sectionを活用することは、ほぼ無い
- **minimal-responsesが必ずしも有利でない用途**
  - メールサーバ等、MXレコードとそれに対応するA/AAAAを大量に引く場合
    - MXに対応するホストのA/AAAAがadditional sectionに入っている場合があり、A/AAAAのDNS検索を省略できる場合があるため

minimal-response **no**なDNSキャッシュサーバでMXを引く➡

```
$ dig dnsops.jp MX
;; ANSWER SECTION:
dnsops.jp. IN MX 10 MX.dnsops.jp.

;; AUTHORITY SECTION:
dnsops.jp.    IN NS    ns1.dnsops.jp.
dnsops.jp.    IN NS    ns2.dnsops.jp.

;; ADDITIONAL SECTION:
MX.dnsops.jp. IN A 210.171.226.61
ns1.dnsops.jp.  IN A  210.171.226.61
ns2.dnsops.jp.  IN A  183.181.160.83
```

Unboundとminimal-responses

- DNSラウンドロビンと一緒にminimal-responsesのパッチも提出
  - これもあっさり取込まれて、正式リリース済！(Unbound 1.4.17)

minimal-responsesはデフォルトでオフなので、オンにするには以下の設定を unbound.confに入れてください

```
server:
    minimal-responses: yes
```

まとめ

- Unboundの高性能が生かせる大規模用途をサポートする機能を実装してみました。
  - DNSラウンドロビンによるサーバ負荷分散や、巨大なRRSetについてはいろいろ議論がありますが、運用されてるのは事実
  - これらをサポートに問題があることがハードルとなりUnboundが利用されないのはもったいない！
- Unboundは高性能で脆弱性も少なめなので、おすすめです。キャッシュDNSサーバを構築する時はぜひ検討してみてください。

RFC3484 Rule9について

クライアントIP 192.168.0.1　11000000 10101000 00000000 00000001

サーバIP1　192.0.2.1　11000000 00000000 00000010 00000001
サーバIP2　203.0.113.1　1100 1011 00000000 01110001 00000001

longest matchな 192.0.2.1が優先

クライアントIP 192.168.0.1　11000000 10101000 00000000 00000001

サーバIP1　203.0.113.1　1100 1011 00000000 01110001 00000001
サーバIP2　203.0.113.2　1100 1011 00000000 01110001 00000010
サーバIP3　203.0.113.3　1100 1011 00000000 01110001 00000011

long matchの長さがIP1,IP2,IP3が同一

➡Rule.9的には引分け