第 XVII 部

DNS extension and operation environment

第 17 部
DNS extension and operation environment

第 1 章　DNS ワーキンググループの活動

DNS ワーキンググループは、おもに DNS に関する情報交換の場として、WIDE 研究会や WIDE 合宿にて、BoF を開催している。本文章では、2005 年中に開催された DNS ワーキンググループ BoF にて話し合われた事項や、議論となった事項をまとめた。

第 2 章　DNS ワーキンググループ BoF にて行われた情報交換ならびに議論のまとめ

2.1 はじめに

本章は、DNS ワーキンググループが開催した BoF において、議論ならびに報告の行われた事項に関して、まとめをおこなったものである。本章は、以下にあげる情報に関してのまとめである。

- DNS anycast サービス運用ガイド
- bind8 cache server の問題点
- bind8 と bind9 のゾーン転送性能評価
- bind9 高速化プロジェクト
- 高機能 DNS ライブラリの設計と開発

2.2 DNS anycast サービス運用ガイド

DNS anycast サービスの運用ガイドを示したインターネットドラフトを作成した。次節に概要を示す。

ガイド作成の動機

- DNS サーバの anycast 運用が増えている
- 運用に関するガイドラインが存在しないので作成したい
- 運用上のミスを減らしたい

前提条件

- ISP から接続性を提供されるモデル
- グローバルに経路を広告するモデル

運用における検討項目

- ISP の選択
  - ISP の外部への接続性
  - サービスセグメントと管理セグメントの分割が可能
  - IPv6 の接続性
- サーバロケーションの選択
  - ロケーションのセキュリティレベル
  - 同じ場所に集中させずにロケーションを多様化
  - 電源の冗長化
- 設置ならびに運用コスト
  - 設置のためのコスト
  - メンテナンスのためのコスト
- 計測ならびに評価の手法
  - ICANN CNNP テスト
  - ルーティングの安定性
  - Reverse Path Forwarding によるフィルタリングへの対処

2.2.1 BGP Anycast Node for Authoritative Name Server Requirements

IP エニーキャスト技術を DNS に適用することによって、ネームサーバのオペレータは、DNS サーバの台数を増強して、地理的に多様な位置に分配することができる。これは DNS のプロトコルにも反しておらず、DNS サーバに対する DoS 攻撃に対する耐性を高めることができ、負荷分散を行うことが可能となる。

しかしながら、IP エニーキャストを用いた DNS サーバでは、すべての DNS サーバがサービスに同じ IP アドレスを用いるため、クライアントがどのエニーキャスト DNS サーバに対して通信を行っているかは特定ができない。

したがって、エニーキャスト DNS サーバの 1 台が、もしなんらかの理由で間違った応答を返すような障害が発生した場合には、ユーザから見て原因の特定が難しくなる場合がある。これは IP エニーキャスト技術の展開のためのリスクの 1 つである。

このインターネット・ドラフトは、DNS サーバにエニーキャスト技術を適用するにあたっての前提条件について述べ、その際に問題となる点や、留意しておくべき共通の事項に関して述べた文章である。

2.3 bind8 cache server の問題点

bind8 を cache server として使う場合にはいくつかの既知の問題点が存在する。その問題点は以下のとおりである。

(1) ある名前に対して UDP に収まりきらない大量の RR が登録されており、かつ TCP 53 番ポートがフィルタされていて到達性が無いと、大量にクエリを送信し続ける

(2) glueless delegation が存在すると、その delegation を引くきっかけとなったもともとのクエリを忘れてしまう

[1] bind8 は、delegation 先の zone 内に glue がないと、検索を止める

[2] 名前解決リクエストを出したクライアントは、待っていても回答が来ないので、リクエストを再送する

[3] bind8 があらためて検索を行う 2 度目の検索は、前回途中まで行った検索の cache が残っているため、最後まで検索が行われる

[4] クライアントから見ると、結果として検索に時間がかかる

(3) bind 8.2.7 までの実装では、glue 無しの delegation が 2 回連続すると、名前の検索が不可能となる

以上の点から、cache server として利用する場合には、bind8 はお薦めできない。

2.4 bind8 と bind9 のゾーン転送性能評価

ゾーンの転送性能評価を JP DNS がもつ 64 個のゾーンを使って行った。評価結果を表 2.1 に示す。

- bind9 → bind{8,9} はそれなりの性能が出ている。
- bind8 → bind9 は転送にかなり時間がかかってしまう

なお、実験は同セグメント接続されている 2 台のサーバにて行った。

表 2.1. ゾーンの転送性能評価

bind	(MB)	(秒)		転送速度 (Mbps)	
version	転送量	平均	標準偏差	平均	標準偏差
9 → 8	33	48.3	7.3	0.70	0.12
9 → 9	33	22.4	1.1	1.47	0.08
8 → 8	85	60.4	12.6	1.46	0.26
8 → 9	85	100.3	14.2	0.87	0.12

2.5 bind9 高速化プロジェクト

目標

- 「bind9 は bind8 より遅い」という定説の打破
- 応答性能は 1 CPU で bind8 比 80%の達成
- 追加 1 CPU あたり 50%増の達成
- 起動時間を bind8 より早く

改善策

- additional section の内部キャッシュ
  bind8 比 60–90%程度に改善
- thread における mutex lock の見直し
  lock を減らして並列性を向上させる

評価環境

- hardware/software
  AMD opteron 2 GHz × 4、RAM 3.5 GB
  GbE NIC
  Freebsd 5.3、suse Linux 9.2、Solaris 10
  bind 9（2005 年 1 月頃の CVS snapshot）、
  bind 8.3.7、nsd 2.2.0
- server configuration
  root server（randam、DoS）
  .net server
  多数の zone を管理する auth server
  10,000 zone、100RR/zone
- cache server（moderate、busy）
  TTL を短くした RR を検索させ、繰り返し検索に行く状態で計測

結果（Linux）

- root server
  bind9: 2000 qps/1 thread、
  70000 qps/4 thread
  bind8: 2000 qps/1 thread、
  30000 qps/4 thread
  nsd: 9000 qps/1〜4 thread
  目標達成
- .net server
  bind9: 30000 qps/1 thread、

80000 qps/4 thread
目標達成

- many zone
  .net とほぼ同じ
  目標達成
- root server に DoS
  bind9: 20000 以下 qps/1 thread、
  30000 弱 qps/4 thread
  bind8: 20000 以下 qps から thread が増えると徐々にさがっていく
  nsd: 70000 qps ぐらい常にキープ、安定している
- cache server 1
  bind9: 20000 qps/1 thread、
  50000 qps/4 thread
- cache server 2
  bind9: 30000 qps/1 thread、
  70000 qps/4 thread

今後の予定

- 一般公開
- multi CPU での評価継続
- zone のメモリイメージをそのまま file にダンプし、次回起動時にはそれを mmap して利用することによる高速化

2.6 高機能 DNS ライブラリの設計と開発

開発の動機

- 古い bind ベースのリゾルバライブラリの呪縛から脱するために新たなリゾルバライブラリを開発する

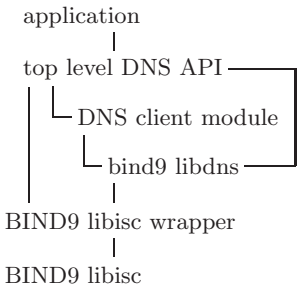
bind ベースリゾルバライブラリの問題点をあげる

- ブロッキング動作を基本としているため、タイムアウトが発生するまで次の名前解決に進まない
- API として DNS format を直接触るようなインタフェースしか用意されていない
- 拡張性に乏しい設計となっており、たとえば DNSSEC のためのインタフェースを追加するなどの拡張が難しい
- bind9 のライブラリは多機能であるが、多機能過ぎて使いにくい

設計の方針

- bind9 のコードを流用し、必要な機能のみを実装した汎用 DNS ライブラリを目指す
- メモリ／イベント管理等の機能は抽象化した wrapper 経由で利用

アーキテクチャ案



- 同期/非同期モードを選択可能
- 名前解決処理が終ると action がコールバックされる
- 結果が namelist に入り、アプリケーションに渡される

実装にあたっての Open issues

- コードサイズ
  リゾルバライブラリとしてはサイズが大きくなってしまう
- DNSSEC interface
  validation に失敗したことをアプリケーションに伝えるために、EAI_INSECUREDATA のような追加コードが必要となる

第 3 章　まとめと今後の方針

2005 年の DNS ワーキンググループ BoF では、オペレーションの観点から見たアイディアの情報交換や、DNS の実装そのものの改善に関する成果の報告が行われた。

今後も DNS ワーキンググループは、グローバルな観点での DNS に関する出来事や情報、さらに DNS の改善のために必要と考えているアイディアに関して、情報交換と議論を行っていく。