2. メールの受信設定の基本

メールを送ってもらうための設定

どうやって送り先を教えるか

- Internet
  - 直接配信

    → DNSに配信先を定義
- バケツリレー・システム
  - UUCPなど(JUNET時代)

    → 経路上の(すべての)ホストに配信先を設定
  - mailconfの活躍

送ってもらったメールの受理

- 届いたメールを自分宛てと認識
  - ローカルに配信(受理)
- 自分宛でないと判断した場合
  - 転送先を探す

DNSレコードの基本型

name [ttl] IN type value...

<左辺> <右辺>

- type
  - レコードの種類 (A, MX, CNAME,....)
- value
  - そのレコードの値
  - レコードの種類ごとに形式が異なる

メール配信時に参照されるもの

- A (Address) RR (Resource Record)
  - ホストのIPアドレスを定義
- MX (Mail eXchanger) RR
  - メールの配信先を定義
- CNAME (Canonical NAME) RR
  - ホストの別名を定義

ホストに対するIPアドレスを定義 (A RR)

- ホスト名　　IN　A　IPアドレス

  mail.x.co.jp.　IN　A　12.34.56.78

- user@mail.x.co.jp 宛てのメールの配信時に参照

DNS登録状況の確認ツール

- nslookup
- dig
- host
- dnsquery
  - 以上、bind に附属
  - できるだけ新しいバージョンを使う

nslookupでAを確認(1)

% nslookup sh.wide.ad.jp.

Server: localhost

Address: 127.0.0.1

Name: *sh.wide.ad.jp*

Address: *203.178.137.73*

複数のIPアドレスを持つホスト

```
mail.x.co.jp        IN  A  12.34.56.78
                    IN  A  12.34.54.32
```

- 最初のアドレスへの配信に失敗した場合、順に全てのアドレスを試みることが期待される(実装依存)
- DNSのラウンドロビン機能により、検索で得られる順序が毎回変わる
  - 負荷分散
  - そのうち届く(?)

nslookupでAを確認(2)

% nslookup jp-gate.wide.ad.jp

Server: localhost

Address: 127.0.0.1

Name: *jp-gate.wide.ad.jp.*

Addresses: *203.178.137.17, 203.178.136.81, 203.178.137.75, 203.178.136.89*

Generic なメールアドレス

- ホスト名部分を持たない
  - ホストの改廃に依存しない
- MX (Mail eXchanger) RR を利用
- メールアドレス IN MX 優先度 ホスト名

  x.co.jp. IN MX 10 mail.x.co.jp.
- user@x.co.jp 宛てのメールは指定されたホストに送られてくる
  - MX を引いてから、右辺について A を引く

nslookupでMXを確認

% nslookup -q=mx wide.ad.jp.

Server: localhost

Address: 127.0.0.1

wide.ad.jp preference = 10, mail exchanger = sh.wide.ad.jp

:

sh.wide.ad.jp internet address = 203.178.137.73

(additional information)

災害に備える(MX編)

- メールの受信の代行

```
x.co.jp.        IN MX   10 mail1.x.co.jp.
                IN MX   50 mail2.x.co.jp.
                IN MX 100 mail.provider.ad.jp.
```

- 数字が小さい程優先度が高い(コスト値)
  - 送り側は成功するまで順にコストの大きなものへ配信を試みる
- mail1 の回復後にまとめて転送
  - メールの保存期間に注意

MXのプレファレンス

- MX RRに指定するコスト値
- コスト最小
  - Primary MX / Primary Mail Server
  - First MX / First Mail Server
- コスト準最小
  - Secondary MX / Secondary Mail Server
- コスト最小以外
  - Lower MX (優先度が低いという意味)

Lower MXの条件

- メール･ループを防ぐための条件
- MX RR の右辺ある自分の名前を認識
  - 自分自身への接続の防止
    - sendmail -bt において $=w で確認
    - インタフェースのアドレス応じた名前は自動登録
- 自分の名前に対する MX RR のプレファレンス以上のコストのRRを捨てる
  - Lower MX 間でのピンポンの防止

負荷分散

```
x.co.jp.        IN MX  10 mail1.x.co.jp.
                IN MX  10 mail2.x.co.jp.
```

- 同じコストの場合は送り側が乱数で選ぶ
- ラウンドロビン機能による負荷分散も可能
  - A RR に架空のホストを定義
  - 複数のホストのIPアドレスを記述
    - spool full などプロトコル的に拒否された場合の動作が MX による場合と異なる
- 最終的に一つのメールボックスへ
  - 受信側の仕掛けも必要

ホストにもMXレコードを

- 災害に備えるため
  - Aレコードだけでは Secondary MXが指定できない
  - 異なるホストに対するIPアドレスを定義したAレコード(仮想ホスト)
    - 負荷分散にしかならない
- DNS検索の効率化
  - そのホストしか受信しなくても定義すべき
    - 一度で検索が完了する (後述)

ワイルドカードMX (cont.)

- *.x.co.jp. IN MX 10 mail.x.co.jp.
- Firewall がある場合(直接通信できない)
  - 外: 個々のレコードを外部に見せたくない
    - でもホスト宛のメールアドレスを利用したい
  - 内: 外界をひとつのレコード定義で代表
    - root に Wildcard MX
- nohost.x.co.jp や host.nosubdom.x.co.jp にマッチ
  - メールが飛ぶ

ワイルドカードMX (cont'd)

- specific なレコードが存在すると参照されない

```
ns.x.co.jp.        IN A  12.34.56.78
*.x.co.jp.         IN MX 10 mail.x.co.jp.
ns.x.co.jp.        IN MX 10 mail.x.co.jp. (必要)
```

  - サブドメインが存在する場合も同様
- ワイルドカードMXの弊害
  - → メールアドレスの補完

CNAME (Canonical NAME) RR

- 他の名前へのエイリアス(別名)
- サービス名をエイリアスとして定義
  - archie.wide.ad.jp (=sun3.tokyo.wide.ad.jp)
- サービスホストの変更に便利
  - pop.x.co.jp. IN CNAME host1.x.co.jp.
- 一つの名前に対してCNAMEを他のRRと一緒に定義してはいけない
  - エイリアスに徹する

CNAME RR (cont'd)

- CNAME RR の左辺に対応するメールアドレスの受理の設定
  - 右辺の名前のホストにメールが送られる
- MX RR への CNAME もあり?
  - 本来は A RR を指すべきもの

nslookupでCNAMEを確認

% nslookup -q=cname archie.wide.ad.jp.

Server: localhost

Address: 127.0.0.1

archie.wide.ad.jp canonical name =
sun3.tokyo.wide.ad.jp

DNS検索の手順 (cont.)

1. CNAMEを解決
   - CNAME でなくなるまでチェインをたどる
     - 上限あり(無限ループ防止)
2. MXで検索
   - 複数あれば、preference でソート
   - MXが見つかった時、Aも同時にAdditional Information として得られる

DNS検索の手順 (cont'd)

3. Aで検索
   - MXが得られなかった時
   - 個々のMXに対して(Additional Info.でAが得られなかった時)

- A しか定義されていなければ、検索処理は2回必要 (MX と A)
  - ホストにも MX を定義すべき
    - 通信トラフィックの削減

その他の注意事項

- ホスト名に利用できる文字
- MX RR の右辺と CNAME
- メールアドレスと CNAME
- CNAME のチェイン

「ホスト名」に利用できる文字

- アルファベット (A-Z, a-z)
- 数字 (0-9)
- ハイフン (-)
- 注意すべき文字
  - アンダースコア (_)
    - RFC1035(S), RFC1123(S)は許していない
    - 新しい(4.9.4以降の)bindのresolverは、_ を含むホスト名を無視する (res_hnok)
      - メールが落ちる

MX RR の右辺と CNAME

- MX RR の右辺に CNAME の左辺となる名前を書くべきではない
- Lower MX が MX RR の右辺にある自分の名前を認識できないと問題
  - 回避策が講じられていれば動くけど…
  - namedが警告を出す

メールアドレスと CNAME

- エンベロープのエイリアスは本名に書き換えられるべき(RFC1123(S))
- 多くの(古い)sendmailはヘッダも本名に書き換えてしまう
  - どのアドレス宛に届いたのかがわからなくなる
  - sendmail.cf の設定次第
- 書き換えられたくないときは、MX か A で
  - IETFはCNAMEによる書き換えをしない方に動いている(?)

CNAME のチェイン

- CNAME RR の右辺がさらに別の CNAME RR の左辺

```
alias1          IN  CNAME  alias2
alias2          IN  CNAME  real-name
```

- RFC1034(S)
  - 定義は非推奨(should not)
  - 実装では辿れること(should)
    - sendmail では10回までたどる (MAXCNAMEDEPTH)
    - named は8回までたどる (MAXCNAMES)

DNS管理入門
- DNSデータの追加作業 -

- ゾーン･ファイルを探す
- レコードの追加
  - 自動生成していないか確認
- シリアルを増やす
- データの再読み込み
- セカンダリ･ネームサーバでの更新

ゾーン(zone)

- データの管理の単位となるドメイン空間
  - my.domain.jp zone = *my.domain.jp
- ドメインの一部が別のゾーンとして別管理されることもある

ゾーン・ファイルを探す

- /etc/named.boot (bind 4.x)

  directory /etc/namedb

  primary my.domain.jp my.domain.zone

- /etc/named.conf (bind 8.x)

  options { directory "/etc/namedb"; };

  zone "my.domain.jp" { file "my.domain.zone"; };

→ /etc/namedb/my.domain.zone

レコードの追加 (cont.)

- 左辺を省略した場合は直前のレコードの左辺が仮定される

```
; $ORIGIN my.domain.jp.（説明のため意図的にコメントです）
@           IN SOA ...
            IN NS  …
; my.domain.jp. に対する定義
a           IN A   ...
            IN MX  ...
; a.my.domain.jp. に対する定義
```

レコードの追加 (cont'd)

- MX/CNAME RR は末尾の . に注意
  - デフォルト・ドメインの補完を抑制
- $ORIGIN がなければ、named.{boot,conf} で指定されるゾーンでの相対
- @ は指定されたゾーンのドメイン名自身に対する定義

シリアルを増やす

- SOA レコードの () 内の最初の数字を各ドメインの管理方針に従って増やす
  - セカンダリNSが存在する場合に重要
    - 更新の有無の判定

```
@           IN SOA ... (
                        1997121703 ; Serial
                        3600
                        ... )
```

データの再読み込み

- SIGHUP でデータベースの読み直し

  # ps aux | grep named

  65 ?? .... named

  # kill -HUP 65

- ndc (name daemon control interface)

  # ndc reload

セカンダリ・ネームサーバ

- ゾーン毎に用意されたバックアップ・サーバ
- 複数のネームサーバによる可用性の向上
  → 災害に備える(DNS編)
  - できるだけ同時に参照不能にならないように
    - できるだけ相互依存部分の少ないところに設置
- プライマリのゾーン・データをコピーして持つ
  - プライマリと同等のデータ提供サービス

セカンダリ側の更新

プライマリからゾーン・データをコピー

- 手動通知
  - 自分で更新作業を行う
  - メールで管理者に連絡し、作業してもらう
- 自動通知
  - named 4.9 以降の機能 (BIND_NOTIFY)
  - プライマリでのSOAの更新を自動通知

手動更新

- FORCED_RELOAD機能
  - SIGHUP を受けるとシリアルをチェック
- バックアップファイルを消してから named の再起動
  - named-xfer で転送がおこなわれる

```
# mv mydomain.zone mydomain.zone.bak
# ndc restart
```

受信設定のまとめ

- 相手に送り先を教えること
  - MX レコードを定義
- 自分宛てだと解釈すること
  - ローカルへの配信 (受理)

別個に設定が必要