DNSのシステム設計

IIJ 山口崇徳

自己紹介

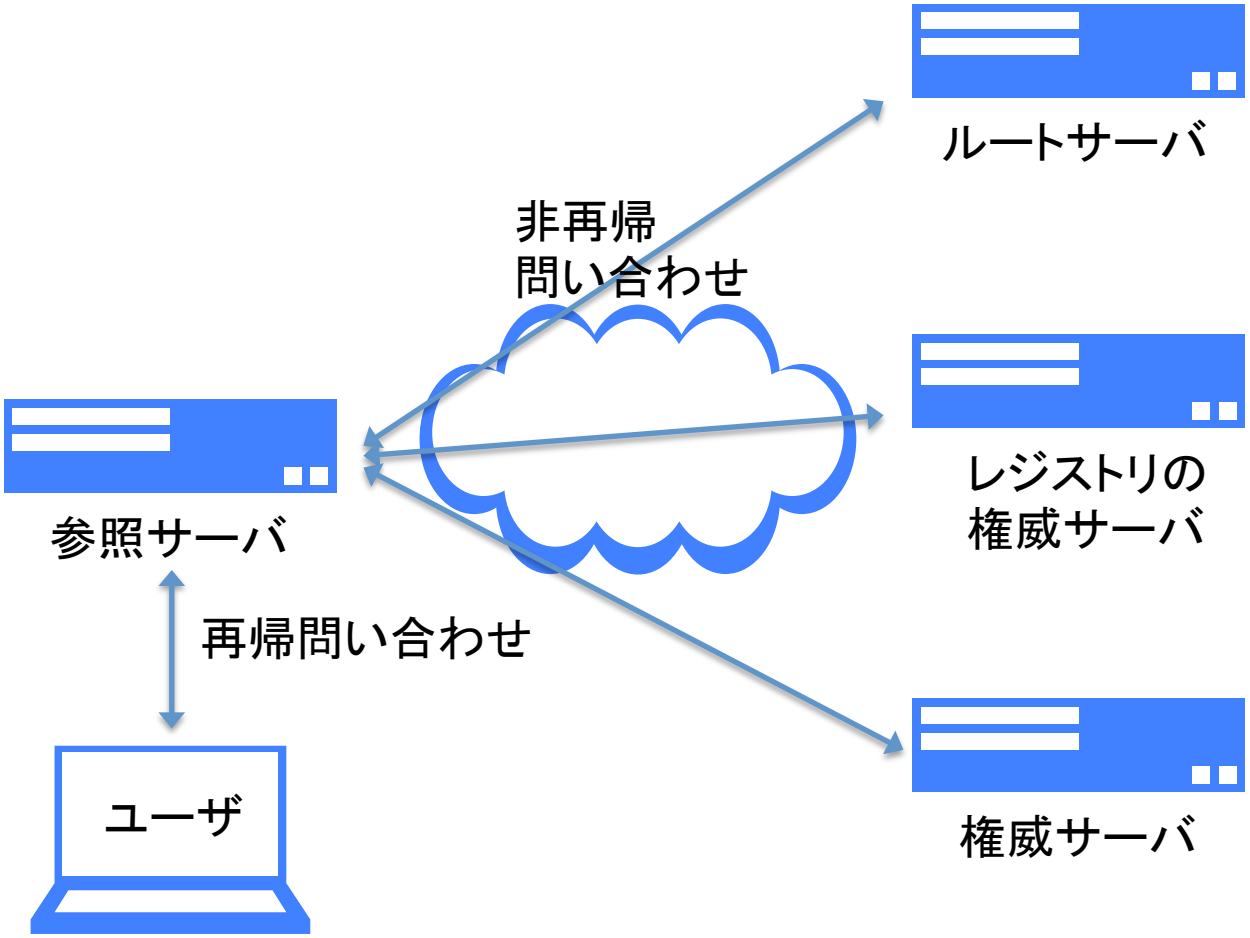
- IIJというところでDNSの運用やってます
  - お客様用参照サーバ
  - お客様のゾーンを預かる権威サーバ
    - DNSSECまわりの開発
  - 某ccTLDのセカンダリ
- 最初のDNSのお仕事は BIND4 → BIND8 の移行
  - 前世紀末
- でも本業はメール屋さん

アジェンダ

- DNS設計の基本
- 権威サーバの設計
- 参照サーバの設計
- 応用

DNS設計のキホン

DNS構成図(超簡略版)



2種類のDNS

- 権威サーバ(authoritative server)
  - コンテンツサーバとも
  - ゾーン情報を管理するサーバ
  - 自分で管理している情報を答える
    - 知らないものを探して答えたりはしない
  - NSレコードに登録するサーバ
- 参照サーバ(recursive server)
  - キャッシュサーバとかフルサービスレゾルバとも
  - 自分ではゾーン情報を管理しない
    - DNSのツリー構造をたどって各地の権威サーバに問い合わせて名前解決し、その結果を答える
  - resolv.confに書いたりDHCPで自動設定されたりするサーバ

役割分担

- 権威サーバと参照サーバは、扱うプロトコルは同じDNSだが、役割がまったく異なる
  - Webサーバ(権威サーバ)とプロキシサーバ(参照サーバ)のような関係
  - 権威サーバは全世界に公開
  - 参照サーバは利用する組織内部だけに公開
  - webと違ってDNSではプロキシ(参照サーバ)の利用が必須という違いがあるが、使い分けとしてはだいたい同じ
- 役割が違うので明確に分離すべし
  - BINDやMicrosoft DNS Serverは両者を混在させることができる
  - できるけど、やらない
  - Apacheだってコンテンツ提供する機能とプロキシの機能をひとつのサーバでいっしょに動かすことはないよね?

なぜ混在させちゃいけないの?

- まったく違う役割のものが同居しているとわかりづらい
  - 権威サーバ(全世界に提供) vs 参照サーバ(組織内部だけに提供)
  - アクセス制限の範囲が違うけどどうやって実現する?
    - BIND はできなくもないが、正しく理解していないと非常に間違いやすい
    - MSDNS ではできない? (すいません触ったことないのでわかりません)
- 権威／キャッシュDNSサーバーの兼用によるDNSポイズニングの危険性について
  - http://jprs.jp/tech/security/2012-07-04-risk-of-auth-and-recurse.html
- http://www.e-ontap.com/dns/weirdra/
- すでに混在しちゃってるサーバがあるなら分離しましょう
  - 参照サーバの機能は残し、権威サーバを新規構築して移行させるのがオススメ

DNSサーバのハードウェア(1)

- スペックは高くなくてよい
  - よほど大きなサイトでなければCPU負荷は小さい
    - DNSSEC有効だと負荷が大きくなるが、それでもたかがしれてる
  - 権威サーバなら、DNSホスティング屋さんのようなよほど大きなゾーンを持つところでないかぎりメモリはほとんど不要
  - 参照サーバはそれなりにメモリを消費するが、それでも数GBあれば十分足りることがほとんど
  - query logを取らなければディスク消費も微々たるもの
  - 帯域は通常は1Mbpsでじゅうぶんお釣りがくる
    - よほど大規模なところでも10Mbpsとか
- パフォーマンスの高さを謳うDNSサーバは多いけど、よほど大きなサイトでもないと、リソースの限界まで使いきるようなことはない

DNSサーバのハードウェア(2)

- ただし、稀にアプリのバグやウイルス感染したPCから全力でクエリを投げられることがあるので注意
  - 最近のPCは性能が高いので、全力勝負すると負けることもある…
  - 毎朝4時のアクセスログ逆引き祭にも注意
  - ふだんはリソースをまったく使わず遊ばせておくぐらいでちょうどいい
- DNSコケたらみなコケる
  - 権威サーバ、参照サーバとも複数指定できるようになってはいるが、1台に障害が発生するとタイムアウト待ちなどで名前解決が遅くなる
  - DNS は障害が起きると他に与える影響が大きいので、堅牢なものを
    - 二重化電源、ホットスワップ可能なディスク、ECCメモリ
    - ロードバランサによる冗長化
    - 仮想化してライブマイグレーションできるように
    - お金はかかるので、どの程度追求するかはよく考えて

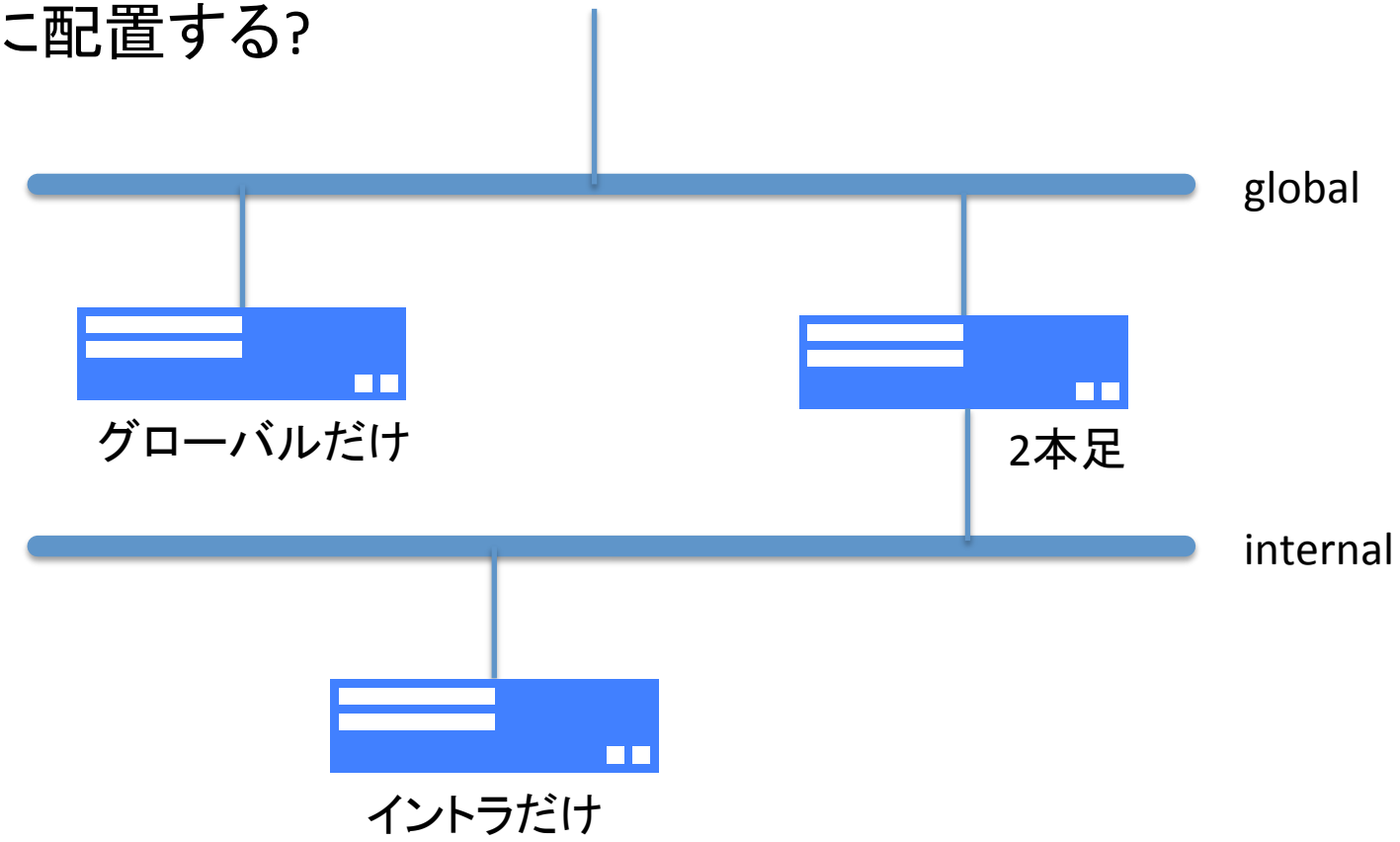
サーバソフトウェアの選択

- 権威サーバ
  - BIND、NSD、PowerDNS、Microsoft DNS、djbdns (tinydns) など
- 参照サーバ
  - BIND、Unbound、PowerDNS recursor、MS DNS、djbdns (dnscache) など
- それを動かす OS の選択
- それぞれの詳細を説明していると時間がいくらあっても足りないので、割愛します
  - 機能、慣れ、メンテの手間などを勘案して判断

- ああ、そういえば7月ですねぇ…

ネットワーク構成(1)

どこに配置する?



ネットワーク構成(2)

- 権威サーバは全世界からアクセスされる
  - グローバル足必須
  - イントラ足が必要かどうかは、メンテなどの際にどこからログインするかで考える
- 参照サーバは組織内部から利用
  - イントラ足必須
  - 外部に名前を問い合わせるためグローバル足も必要
    - グローバル側からの問い合わせを受ける必要があるかは要検討
- グローバル足が不要なDNSサーバ
  - 内部ネットワーク専用の権威サーバ
    - プライベートアドレスの逆引き用など
  - NAT経由でインターネットと疎通があるなら、参照サーバのグローバル足を省略することもできなくはないが…

NAT 利用時の留意点

- NAT テーブルがあふれないように注意
- 実は NAT ではなく NAPT (IPアドレスだけでなくポート番号も変換する)
  - 最近の参照サーバはソースポートをランダム化することでキャッシュ汚染をされにくくしている
  - が、NAT 機器がパケットのソースポートを連番に書き換えてしまうことがある
    - 推測困難なポート番号から容易に推測できる番号に
  - ポート番号の推測が的中すれば外部からのパケットはNATを越えられる
  - キャッシュに毒入れされやすくなる
- 事前に NAT 機器の仕様を調べておくべし
  - むしろ NAT を使わない構成を推奨

ネットワークの確認

- DNS は UDP だけでなく TCP も使う
- DNS は 512 バイト以上の UDP パケットも使う
  - UDP パケットがフラグメントすることもある
- ファイアウォールその他のネットワーク機器がこれらのパケットをちゃんと扱えることを確認しておく
  - EDNS0 を使わない設定にすれば512バイト以上のUDPパケットを扱うことはないはずだけれど
  - その場合でもTCPが通るようにする必要はある

サーバ監視

- ちゃんと監視しましょうね
  - WebやらメールやらDBやらと考えかたが大きく異なるわけではない
  - nagios その他、ふだんから使い慣れているものでよい
- UDPのトラフィックを計測するとよい
  - DNS 専用のサーバならほぼ UDP のパケット数 ≒ DNS のパケット数
  - TCP の DNS パケットもあるし、DNS 以外の UDP パケットも存在しないわけではないので厳密な値ではないが、傾向を掴むには十分
- もちろん、TCP もしっかり監視を
- DSC (DNS STATISTICS COLLECTOR)
  - http://dns.measurement-factory.com/tools/dsc/
  - DNS に特化した統計情報取得・グラフ作成ツール
  - 詳細な情報を取得できる

権威サーバの設計

名前空間の設計(1)

- どんな名前を使うか
  - http://www.example.jp/ か、ゾーン頂点の http://example.jp/ か
  - http://www.example.jp/foo/ か http://foo.example.jp/ か
  - 子会社は child.example.co.jp か example-child.co.jp か
  - 期間限定のキャンペーンをサブドメインでやるか、専用のドメインを取得するか
    - キャンペーン終了後、取得した専用ドメインはどう扱うか?
  - web 方面では cookie の有効範囲を制御するために、まったく異なるドメイン名を取得して併用することがある
    - yahoo.co.jp が使っている yimg.jp など
    - cookie の仕様がおかしいと思う…

名前空間の設計(2)

- どうやって管理するか
  - foo.example.jp も bar.example.jp も単一の example.jp ゾーンで管理するか
  - foo.example.jp も bar.example.jp も example.jp から切り出して別ゾーンとして管理するか
  - foo.example.jp だけを切り出して別ゾーンにするか
  - 部署ごとにサブドメインを作って、委譲するとか
  - 実験に使う名前空間をサブドメインに分離するとか
  - GSLB で負荷分散する名前だけ別ゾーンに切り出すとか
    - GSLB = global server load balance (広域負荷分散)

名前空間の設計(3)

- どうするかはそれぞれのサイトのポリシーによる
  - どのようなポリシーで運用するかをまず決定しなければならない
- 一般的には
  - ゾーンの数が少ない方が運用が楽
  - ゾーン内のレコード数が少ない方が運用が楽
- が、運用が楽かどうかより、まず「何をしたいのか」ありきでポリシーを決めたほうがよい

- 決定したポリシーにしたがって、権威サーバにゾーンを設置し、必要に応じてサブドメインに委譲する

内部名か外部名か(1)

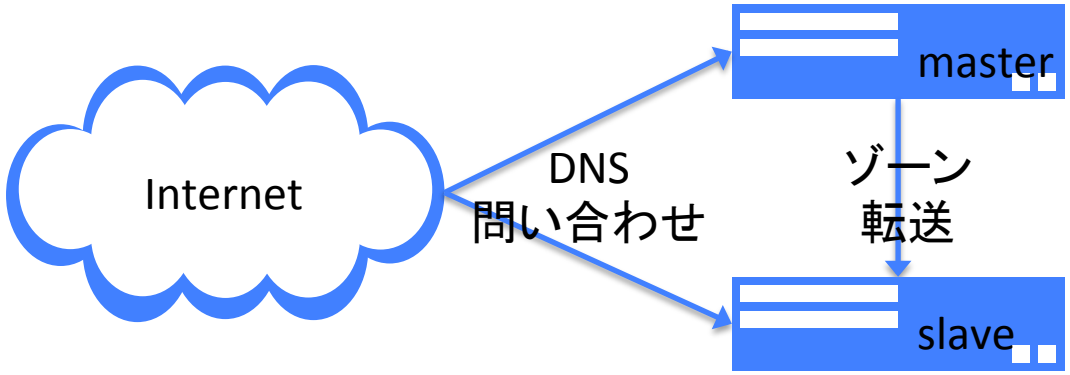
```
example.jp.   IN   NS   ns1.example.jp.    ; 内部名
              IN   NS   ns2.example.com.   ; 外部名
```

- ns2.example.com は example.jp とは異なるゾーンにある名前 = 外部名
- example.jp ゾーンの権威を example.com に預けることになる
  - 万が一 example.com が乗っ取られると、example.jp まで乗っ取られる危険がある
  - example.com の運用は信用できる?
- example.jp の名前解決をするのに、ns2.example.com の名前解決も必要になる
  - 名前解決にかかる負荷、時間が増す
  - これが多段になると、古い BIND8 などでは名前解決不能になることがある

内部名か外部名か(2)

- 可能ならば内部名の方が望ましい
  - NS だけでなく、MX や CNAME なども同様
  - 外部名がダメというわけではない
- どうすればいい?
  - ns2.example.com と同じ IP アドレスを持つ A レコードを内部名 (ns2.example.jp) として登録し、それを NS に登録する
  - ただし、example.com 側の事情で ns2.example.com の IP アドレスが変わるときなどに手間がかかる
    - 外部名なら ns2.example.com の変更に自然に追従する
    - 内部名なら ns2.example.jp を書き換えないといけない

権威サーバの構成




```
example.jp.   IN   NS   master.example.jp.
              IN   NS   slave.example.jp.
```

- マスター1台+スレーブ1台(or それ以上)の構成が一般的
  - マスターでゾーンファイルを管理し、スレーブに転送
  - どちらもNSレコードに載せてインターネットからの問い合わせを受ける
- ネットワーク障害時の影響を小さくするため、2台のホストは異なるネットワークに置くのがよい

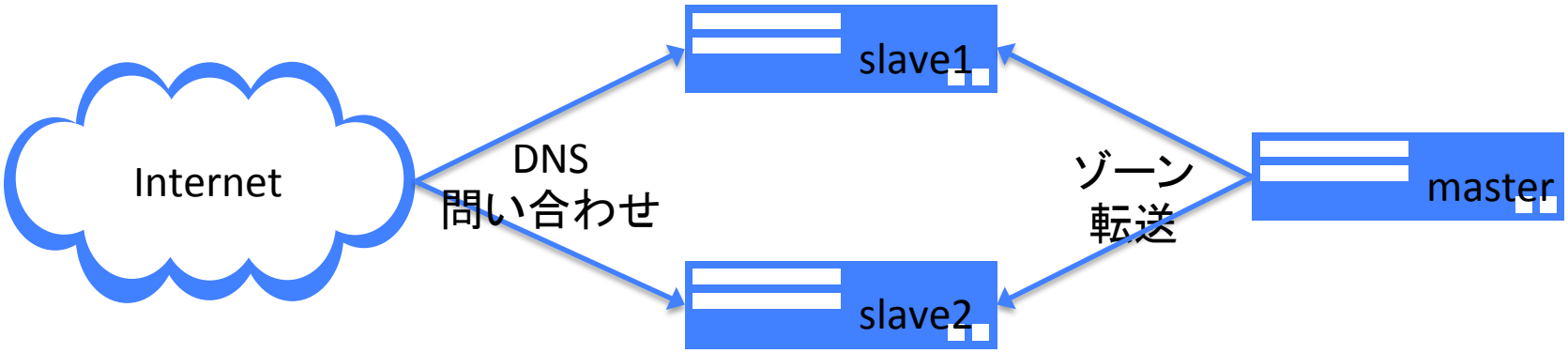
マスターとスレーブの違い

- マスター: ゾーン情報の原本を持つサーバ
- スレーブ: マスターからコピーされたゾーンを持つサーバ
  - キャッシュではない
- マスターとスレーブの違いはゾーン転送の主従
  - あくまで内部的な違いだけ
  - 外部から問い合わせるホストからはどちらも同じように見える
- よくある誤解: マスターが障害で応答しないときだけスレーブに問い合わせる
  - 大嘘
  - どちらにも同じように使われます

NOTIFY

- 大昔のDNSは、マスターのゾーン情報が更新されたかどうかをスレーブが定期的にチェックしていた
  - マスターがゾーンを更新しても、スレーブが更新されるまでにはタイムラグがあった
- 今は、マスターがスレーブに対して能動的に更新を通知する
  → NOTIFY (RFC1996)
  - マスターがゾーンを更新すると、即座にスレーブも更新が完了する
  - NOTIFY を取りこぼす可能性もあるので、旧来の仕組みも併用
- この仕組みが正しく動作するように、マスターサーバでは NOTIFY の通知先(スレーブサーバのアドレス)を設定する
  - スレーブ側は NOTIFY を許可するアドレスを設定(allow-notify)

隠れマスター(hidden master)




```
example.jp.   IN   NS   slave1.example.jp.
              IN   NS   slave2.example.jp.
```

- NSレコードに載せないDNSマスターサーバ
  - 問い合わせを受けるのはスレーブの役目
  - マスターはゾーンの管理とスレーブへのゾーン転送だけに専念
    - DNSSECではさらに鍵と署名の管理
    - インターネットとの疎通はなくてもよい

hidden masterのメリット

- サーバの役割分担の明確化
  - マスター: ゾーンの管理、DNSSECの鍵と署名の管理
  - スレーブ: DNS問い合わせの受けつけ
    - スレーブにはDNSSEC秘密鍵を置く必要がない = 万が一スレーブに不正侵入されても外部からゾーンの改竄を検知できる
- BINDは機能が豊富だけどセキュリティホール多いよ…
  - 外部と接触しないマスターはBINDを使い、スレーブはNSDにまかせる、といった構成が可能
- もちろん通常のマスター/スレーブ構成がダメというわけではない

参照サーバの設計

参照サーバのIPアドレス

- DNS = ホスト名と IP アドレスの結びつきを調べる仕組み
  - IP アドレスを変えても DNS の方を変更すれば同じホスト名でアクセスできる
- が、参照 DNS サーバだけはホスト名で指定できない
  - 参照サーバの IP アドレスを変更する場合には、その参照サーバを利用しているクライアントすべての設定変更が必要
- DHCP や IPCP (IPV6CP)で DNS 設定を配布する仕組みもあるけど…
  - IPCP: PPP 接続でクライアントに IP アドレスを割り当てるプロトコル
  - 管理者の意向を無視して、自動設定されるアドレスを使わず手で設定してあるクライアントがかならずどこかに存在するんだよな…

resolv.confの変更が反映されない

- すっげー頑張って参照サーバの IP アドレスを変更して、/etc/resolv.conf もぜんぶ修正したよ!
  - なのに変更前のアドレスへの問い合わせが止まらない…
- DNS の問い合わせをするたびに resolv.conf が読まれるわけではない
  - プロセス起動直後の初期化時だけ
  - 再初期化しないと resolv.conf の変更は反映されない
  - どうやって再初期化するの?
    - ものにもよるが、プロセス再起動するのが確実
- これに気づかずに変更前の IP アドレスの参照サーバを停止すると名前解決できなくなる

参照サーバのネットワーク設計

- …てなわけで、参照サーバの IP アドレス変更は非常に困難をともなう
- IP アドレスは変更できないものと諦めるべし
  - 参照サーバの稼動前、設計の時点でどのアドレスを使うかを入念に検討しておく
  - ホスト機材のリプレースをおこなうときは同じ IP アドレスを引き継ぐ
- ホストルーティングで引っ越すなんて荒技も…
  - /24 などのネットワーク単位でルーティングするのではなく、IP アドレス1個単位(/32)で無理矢理トラフィックをねじまげる
  - この場合も、別セグメントへの引っ越しはできても IP アドレスが変わるわけではない
- それでもどうしてもアドレスを変更したいときは
  - 地道にがんばるしかないです…

参照サーバの配置

- 複数のネットワークを持つ組織
  - 本社と地方拠点とか
- どこに参照サーバを置くべきか?
  - 全拠点で同じ参照サーバを使う?
  - 拠点ごとに参照サーバを設置する?
    - 人数が少ない拠点だから ISP 提供のものを使っちゃう?
- 拠点ごとに置いた方がいいかなぁ
  - サーバ障害、ネットワーク障害、IPアドレスを変更できない呪いの影響を局所化する
  - 拠点が多くなると管理がめんどうになるけど…
    - 夏の BIND 脆弱性祭への参加とか
- 集中管理することのメリットもあるので、実際どうするかは事前によく検討する

参照サーバのアクセスの偏り

- 参照サーバを複数台用意してるのに、アクセスされるのは1台だけで2台目がほとんど使われない
- デフォルトではクライアントに設定された順にアクセスされる
  - ので、後の方に書いた参照サーバはほとんど使われない
  - resolv.conf に "options rotate" と書くと、参照サーバを順繰りに使うようになってアクセスが分散される
    - …ものもある。Solaris や Linux はそのように動くが、*BSDは非対応
- ロードバランサなどで負荷分散するのが確実
  - でもコストが…

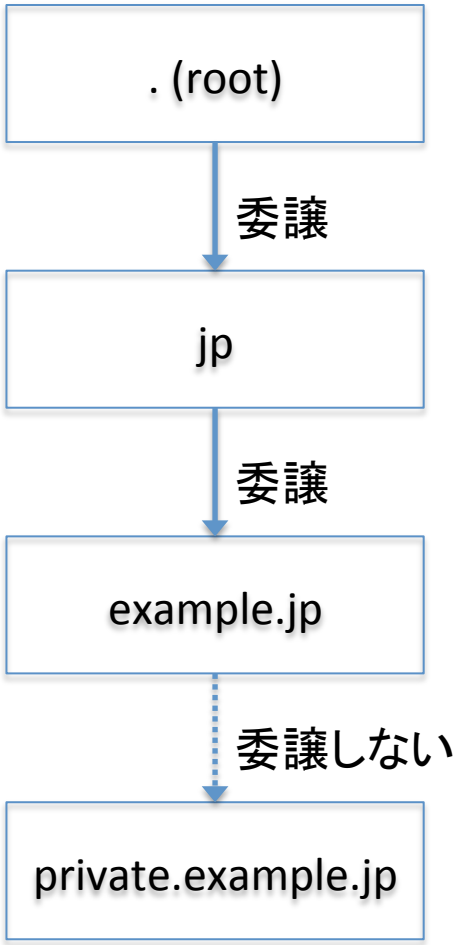
応用

イントラ内部だけのゾーン

- 内部ネットワークのホスト名解決用権威サーバ
- プライベートアドレスの逆引きゾーンは必須
  - ないとどうなるの?
    - プライベートアドレスの逆引きを調べるのに、インターネットにクエリが出ていってしまう
    - 障害でインターネットとの疎通がなくなると、インターネットとは無関係なイントラ内のホストに対するアクセスも困難になる(ことがある)
  - 逆引きが不要でも、中身が空のゾーンを設置すべし
    - SOAとNSレコードだけのゾーン
    - 最近の参照サーバはビルトインで空ゾーンを持っていることが多い
    - 空でない応答を返すならビルトインのものを使わないように設定して、別途ゾーンを定義する
- こういう内部専用のゾーンはどうやって名前解決する?

内部専用ゾーンの名前解決(1)

- 権威サーバはルートサーバから辿って探すのが基本
- NS をどこからも委譲されず、ルートサーバから辿れないところにあるイントラ用権威サーバを使うにはどうすればいい?
  - 右図の例では、private.example.jp をふつーに委譲しちゃうという方法もできなくはない
  - が、168.192.in-addr.arpa を自分のところに委譲してもらうことはできない



内部専用ゾーンの名前解決(2)

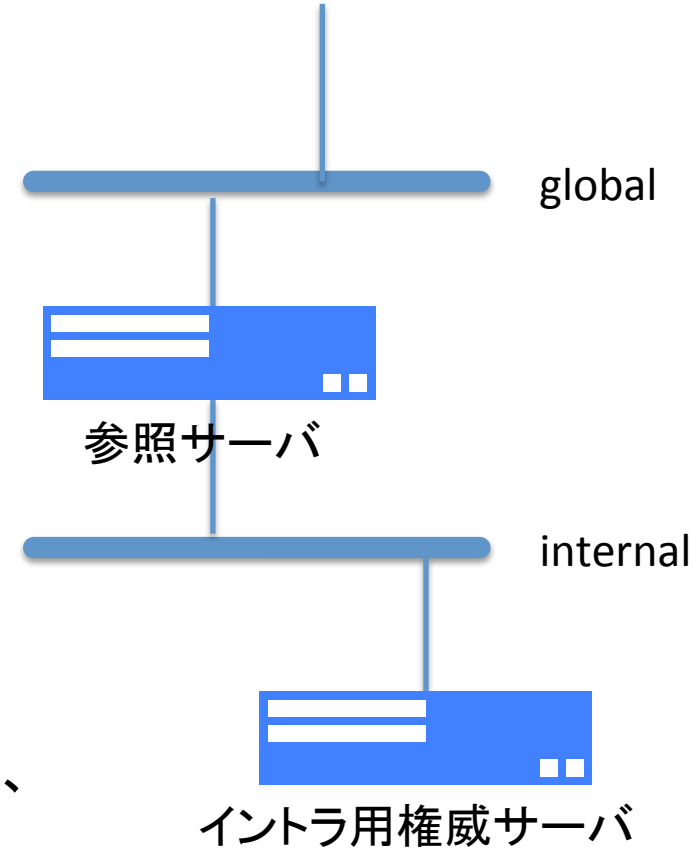
- 参照サーバの設定で内部の権威サーバの場所を教えてやればよい
  - BIND の場合

```
zone "private.example.jp" {
    type stub;
    masters { 権威サーバ; };
};
```

  - Unbound の場合

```
stub-zone:
    name: "private.example.jp"
    stub-addr: 権威サーバ
```

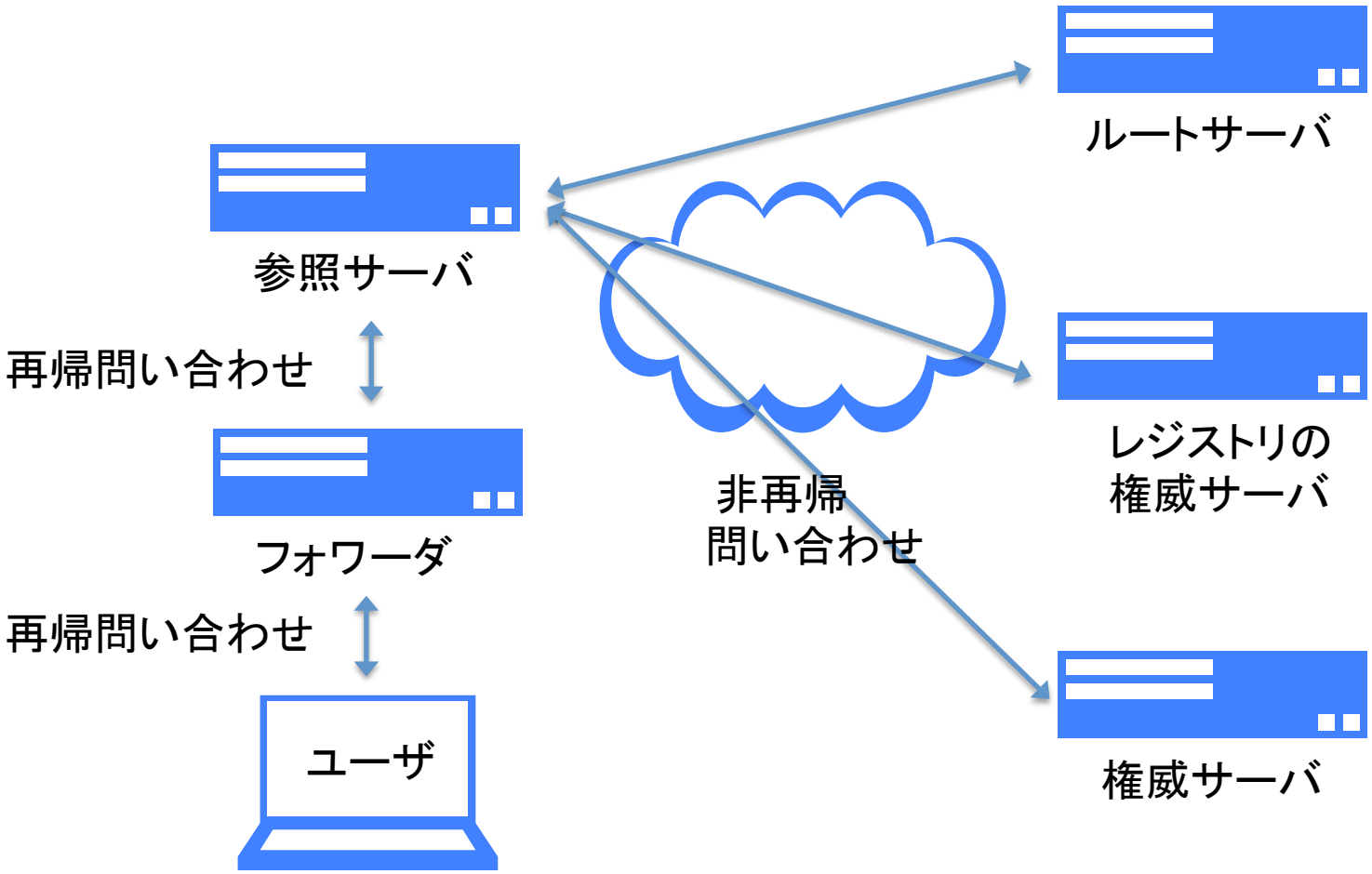
  - 設定したゾーンに対する問い合わせは、ルートサーバからNSを辿って再帰検索するのではなく、指定されたホストに直接問い合わせる



フォワーダ(1)

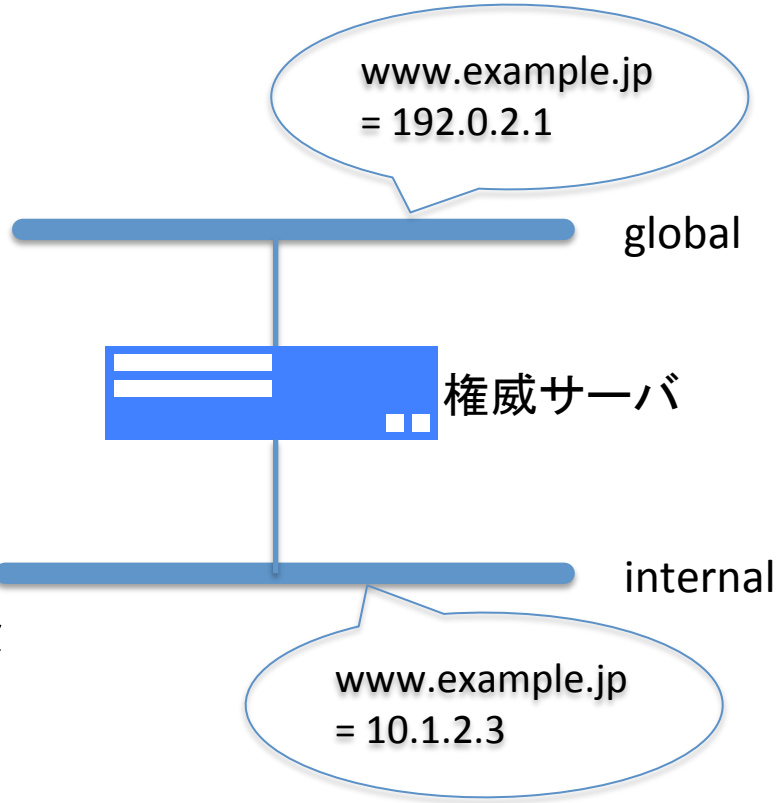
- 先ほどの例は type forward (Unbound では forward-zone) として設定してもほぼ同じことが実現できる
  - が、type forward は再帰検索を要求する(RD=1)
    - 権威サーバではなく、参照サーバへの問い合わせのお作法
  - 権威サーバへの問い合わせは非再帰検索(RD=0)が正しいやり方
    - なので、forwardではなくstubとして設定するのが正しいと思うんですが…
    - BIND の公式マニュアルに「stub は新しい設定では推奨されない」と書いてあったりする
- 自前でできない/やりたくない名前解決をよそに回送(forward)する参照サーバ = フォワーダ
  - グローバルへの疎通のない参照サーバが疎通のある参照サーバに問い合わせをフォワード
  - 再帰検索は遅いので、キャッシュが溜まってる別サーバにフォワード
  - 家庭用ルータのDNS機能はたいていフォワーダ

フォワーダ(2)


ルートサーバ
参照サーバ
再帰問い合わせ
レジストリの
権威サーバ
非再帰
問い合わせ
フォワーダ
再帰問い合わせ
ユーザ
権威サーバ

split DNS

- 問い合わせ元のアドレスによって異なる応答を返す
- BIND では view を使えばよい
- BIND 以外の実装で view 相当の機能を持ったものはない
  - …と思う
  - 1台のサーバでがんばらず、複数の IP アドレスを使って内容の異なるゾーンを持った権威サーバを動かせばよい
  - BIND を使う場合でも、後で妙なしがらみにならないよう、可能なかぎり view は使わず済ますのが無難



まとめ

- DNS サーバを構築するにあたって留意すべき点をいくつか挙げてみました
- DNS サーバそのものより、ネットワーク構成をどうするかとか、ポリシー設計とか、そういう周辺のことの方が重要
  - 動きはじめてからでは手遅れのことがあるので、事前に入念に検討してください
  - そうしないと後で泣くよ
    - 泣いてるよ今…