DNS浸透の都市伝説を斬る
～ランチのおともにDNS～

2011年11月30日
Internet Week 2011 ランチセミナー
株式会社日本レジストリサービス（JPRS）
森下泰宏（オレンジ）・民田雅人（みんみん）

本日の内容

- 浸透問題とは何か
- サーバーの引っ越しと浸透問題
  - 浸透問題が起こらない（正しい）引っ越し方法
  - 浸透問題が起こりうる引っ越し方法
- 浸透問題の正体
- まとめとおすすめ

巷のつぶやき

ISPのWebサイトにも…
（顧客向けFAQや技術解説から抜粋）

- DNSの書き換えを行ったからといって世界中に瞬時にその情報が行き渡る訳ではありません。通常、1週間から2週間の時間(**プロパゲーション**期間)をかけて新しい情報が世界中のDNSサーバーへ**浸透**していきます。
- DNS情報の変更後、新しい情報がインターネット上に**浸透（伝播）**するまでに早い所で数時間、遅い所になりますと数週間かかる場合があります。
- 徐々に徐々に、**水が染みこんでいくように**順番に切り替わって行きます。これを「DNSの**浸透**」と言います。
- …他にも多数存在

海外でも…

- Google検索の結果
  - 「DNS penetration（＝浸透）」約 **2,380,000** 件
  - 「DNS propagation（＝伝播）」約 **1,090,000** 件
- 検索で上位に出たページタイトルの例（＆超訳）
  - Penetration Test DNS（**DNS浸透テスト**）
  - DNS Propagation Checker（**DNS伝播チェッカー**）
  - How to Speed Up DNS Propagation - Technology Tips & Tricks（**DNS伝播をスピードアップ**するには）

どんな時に使われているか

- ゾーンデータの変更の際「新しいデータがインターネット全体に反映されるまでに**時間を要すること**」を説明するために使われているっぽい
- つまり、**言い訳**？
  - 顧客に文句を言われている？

でも、**何か変**じゃね？

- みんな「**新しいデータ**」にばかり注目している
- しかしDNSでは「**古いデータ**」に注目すべき！
  - それはなぜか？

浸透問題の本質

- キャッシュDNSサーバーは**古いデータがキャッシュから消えない限り**、新しいデータを能動的に取りに行くことは**決してない**
- つまり、浸透問題とは「新しいデータが反映されない問題」なのではなく、何らかの理由により「**消えるはずの古いデータが残り続けてしまう問題**」である
- DNSのキャッシュは経路制御（BGP）などとはデータの取り扱いが異なることにも注意
  - DNSには本来「浸透」や「伝搬」といった概念は存在しない
  - BGPでは新しいデータの「伝播」や「浸透」で問題ない

どうして古いデータが残るのか？

1. 正しい方法で作業し、古いデータが消えるのを待っている状態
   - 今回取り上げる浸透問題の範疇（はんちゅう）外
   - 厳密にはこの状態は「浸透待ち」ではない
   - 「新しいデータの浸透（伝播）待ち」ではなく、「古いデータの消滅待ち」と言うのが正しい
2. **正しくない方法**で作業しているため、**古いデータが残ったままになってしまう**状態
   - このことを「DNSが浸透しない」と称している人々（業者含む）が数多く存在している
   - 今回取り上げる浸透問題の**本質**

どんな時に問題になるか？

- **権威DNSサーバーの引っ越し**（NSの変更）を伴う場合に浸透問題が多く発生
  - サービスプロバイダを変更する場合など

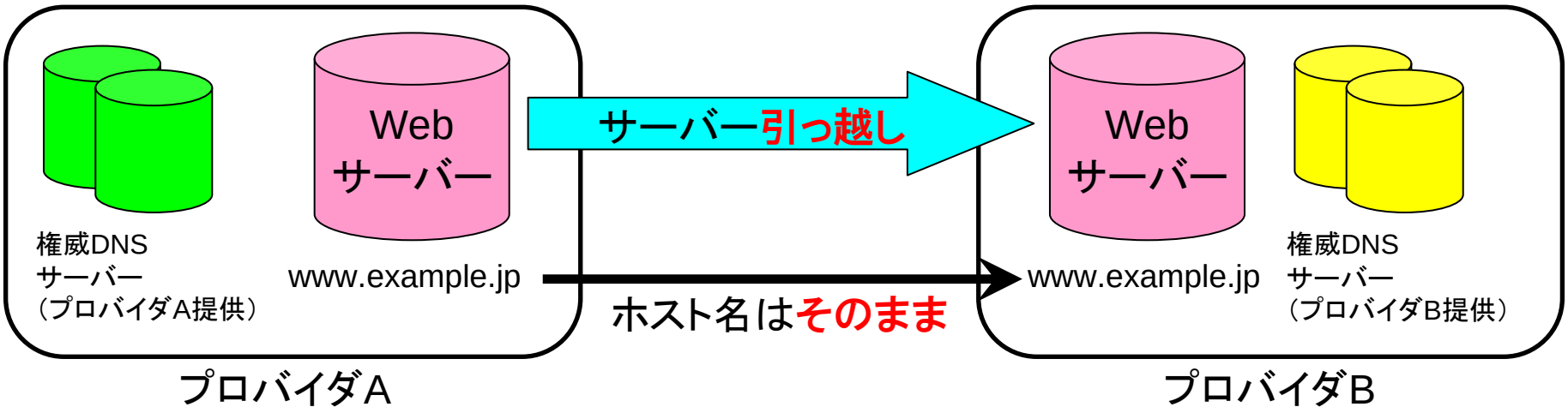
→ 以降ではこの問題に注目します

サーバーの引っ越しと浸透問題

浸透問題が起こらない（正しい）
引っ越し方法と
起こりうる引っ越し方法

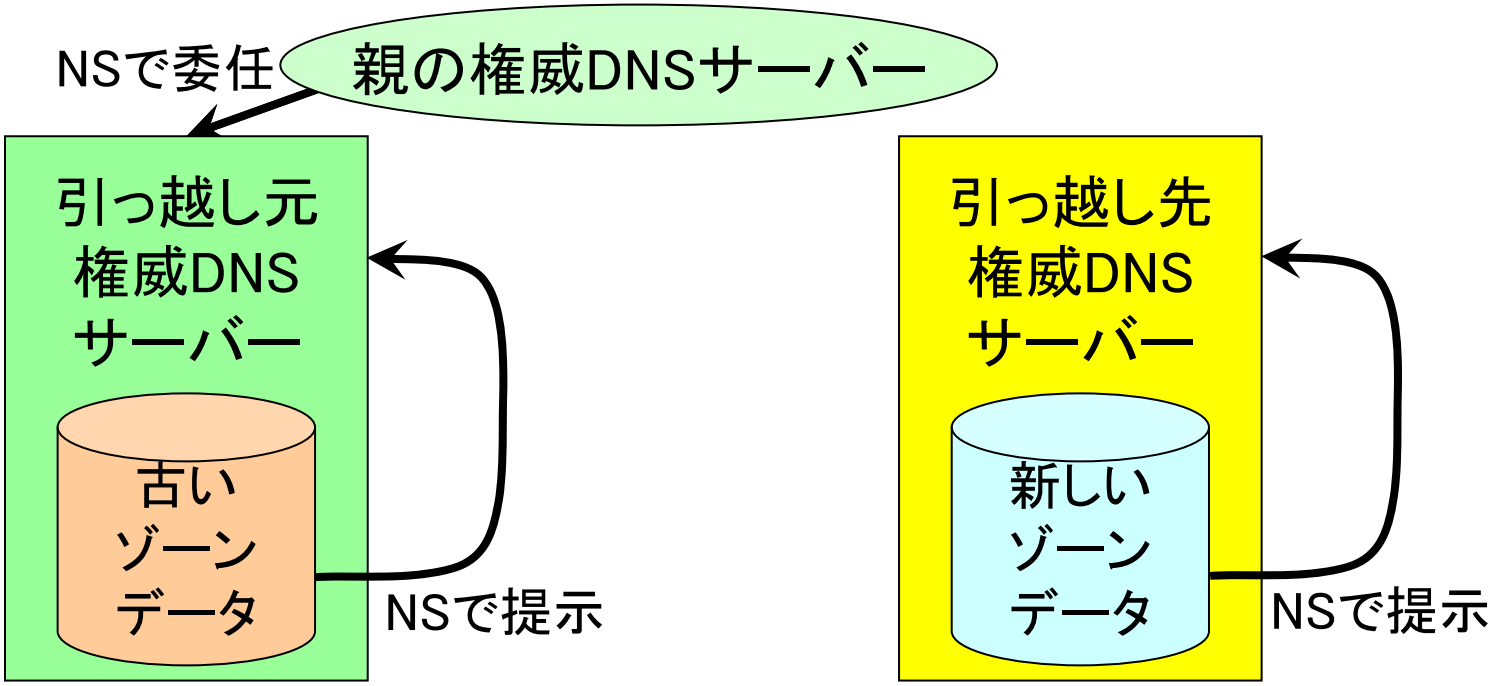
よくある引っ越しの例（プロバイダの変更）

- すべての権威DNSサーバーのホスト名とIPアドレスが変更される
- Webサーバーやメールサーバーなど、権威DNSサーバー以外のサーバーのホスト名は変更されず、IPアドレスのみが変更される



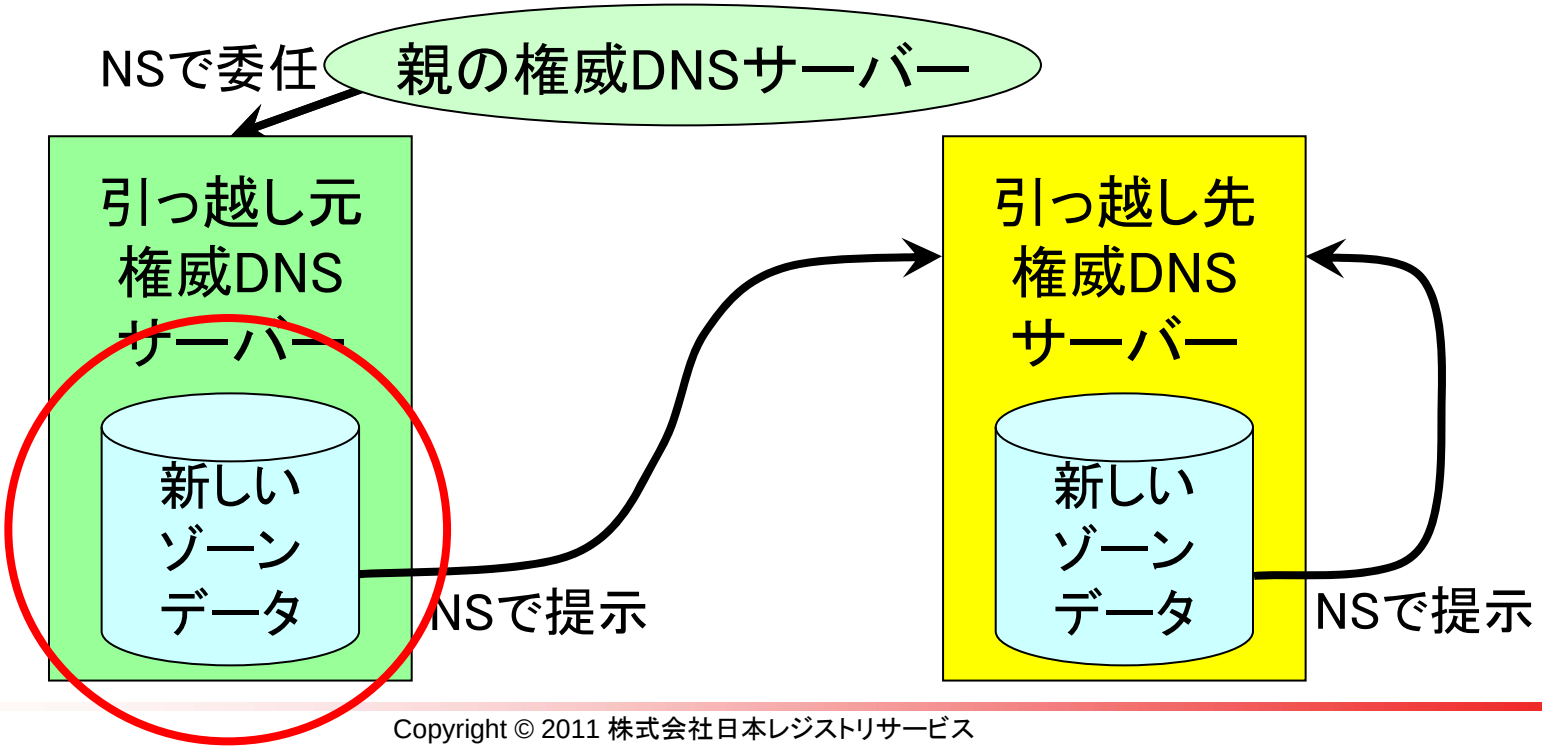
1.引っ越し先の権威DNSサーバーの構築

- 引っ越し先の新しいDNSデータ、新しいNSを設定する
- 引っ越し先のWebサーバーなどもこの時点で作っておく
- 前準備として引っ越し元のAのTTLを短くしておくとよい



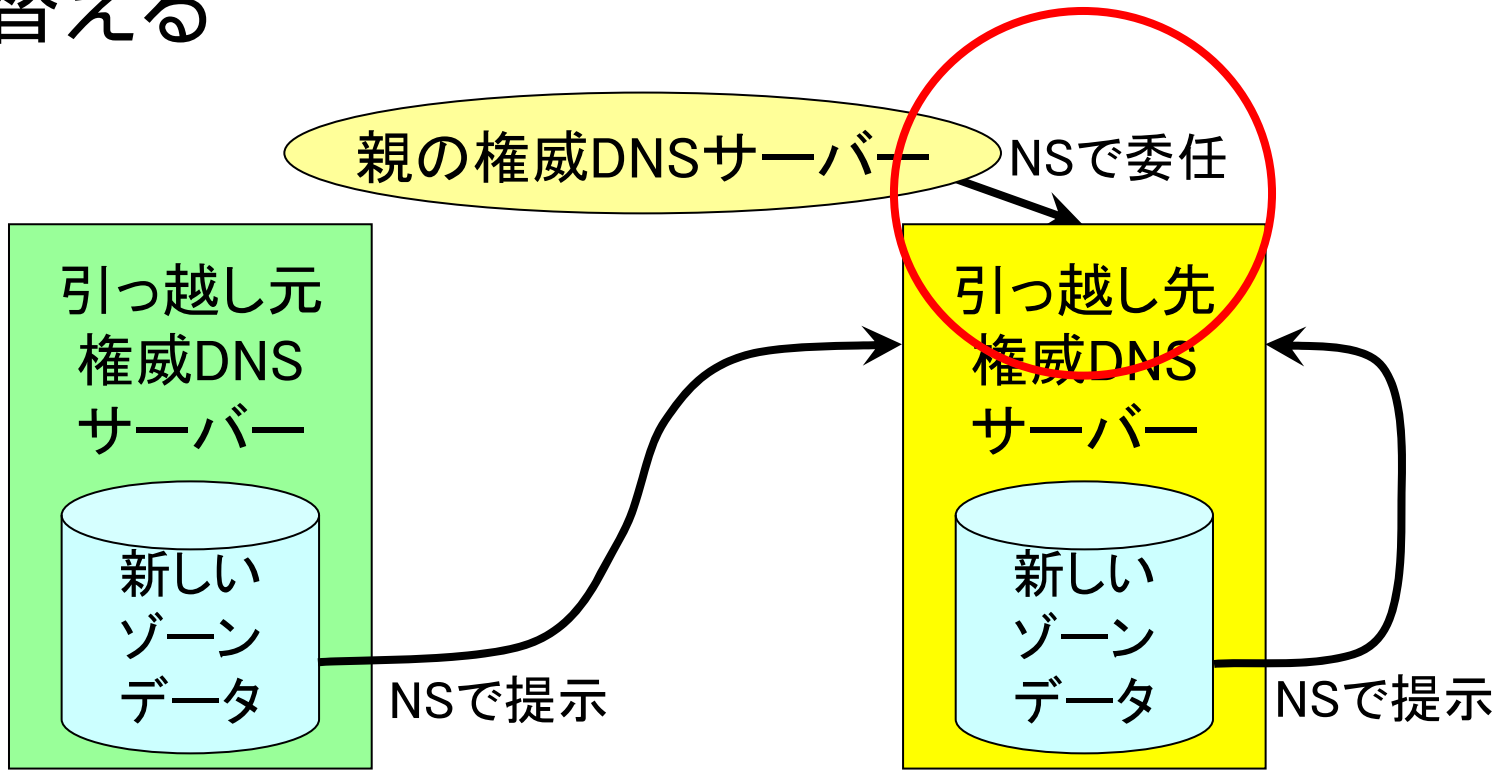
2.引っ越し元ゾーンデータの切り替え

- 引っ越し元の権威DNSサーバーのゾーンデータを、**新しいゾーンデータ(引っ越し先のデータ)**に切り替える
  - NSやグルーも含め**中身**を全部切り替える
  - 新しいゾーンデータのAのTTLは通常の長さで問題ない



3.親に登録したNSの切り替え

- 委任情報（NS、必要に応じてグルー）の変更を親に申請し、引っ越し先の権威DNSサーバーに切り替える

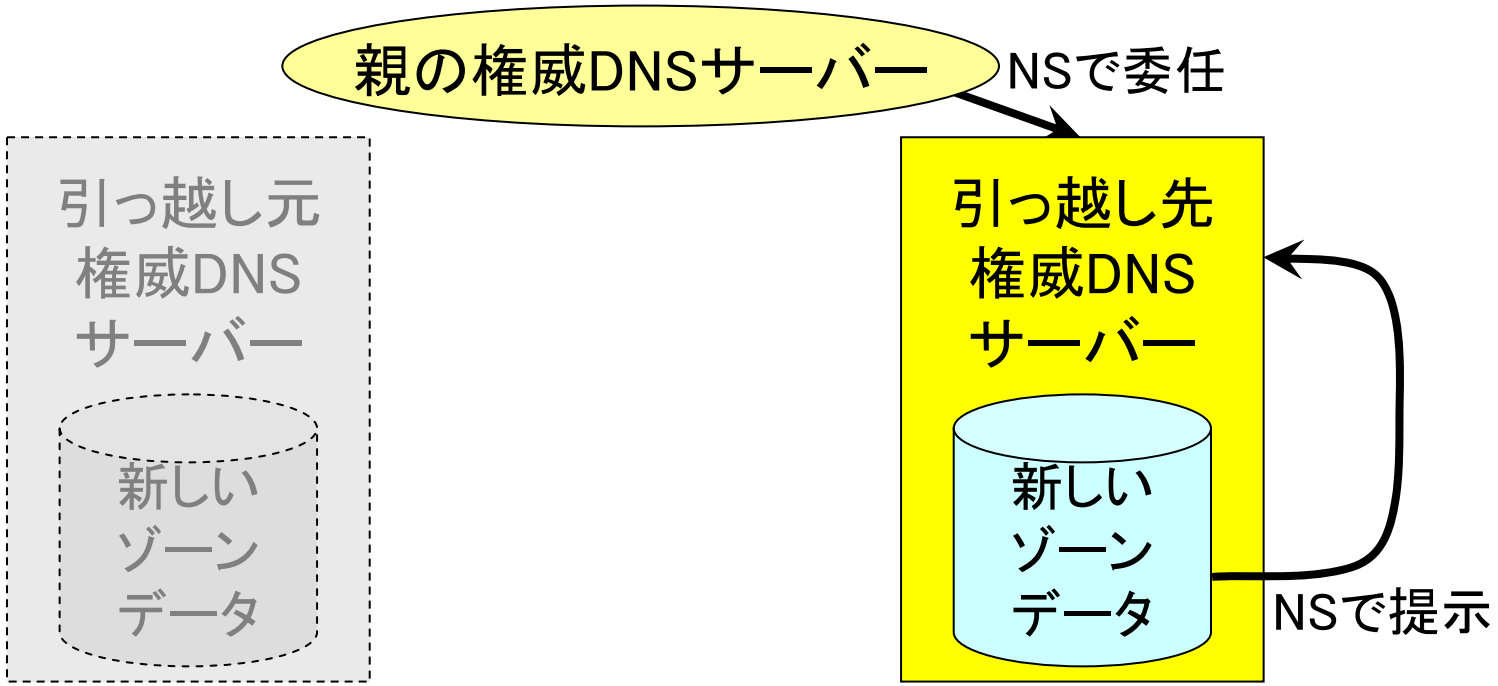


4.この状態で並行運用（古いデータの消滅待ち）

- 以下の**双方**の時間が経過するまで並行運用
  - 引っ越し元権威DNSサーバーのデータを切り替えた時点（手順2）から起算した、引っ越し元権威DNSサーバーのNSで指定していたTTL値（**子の古いNSのTTLの満了**）
  - 親におけるNSの切り替え完了時点（手順3）から起算した、親の権威DNSサーバーのNSで指定されていたTTL値（**親のNSのTTLの満了**）
- 切り替える前のwwwなどのAのTTLを、NSのTTLより短くしてあることが前提

5.引っ越し元の権威DNSサーバーの停止

- 双方のTTL満了後、引っ越し元権威DNSサーバーを停止する
- 停止しなくても実害はない
  - 正しい方法では古いゾーンデータは公開されない

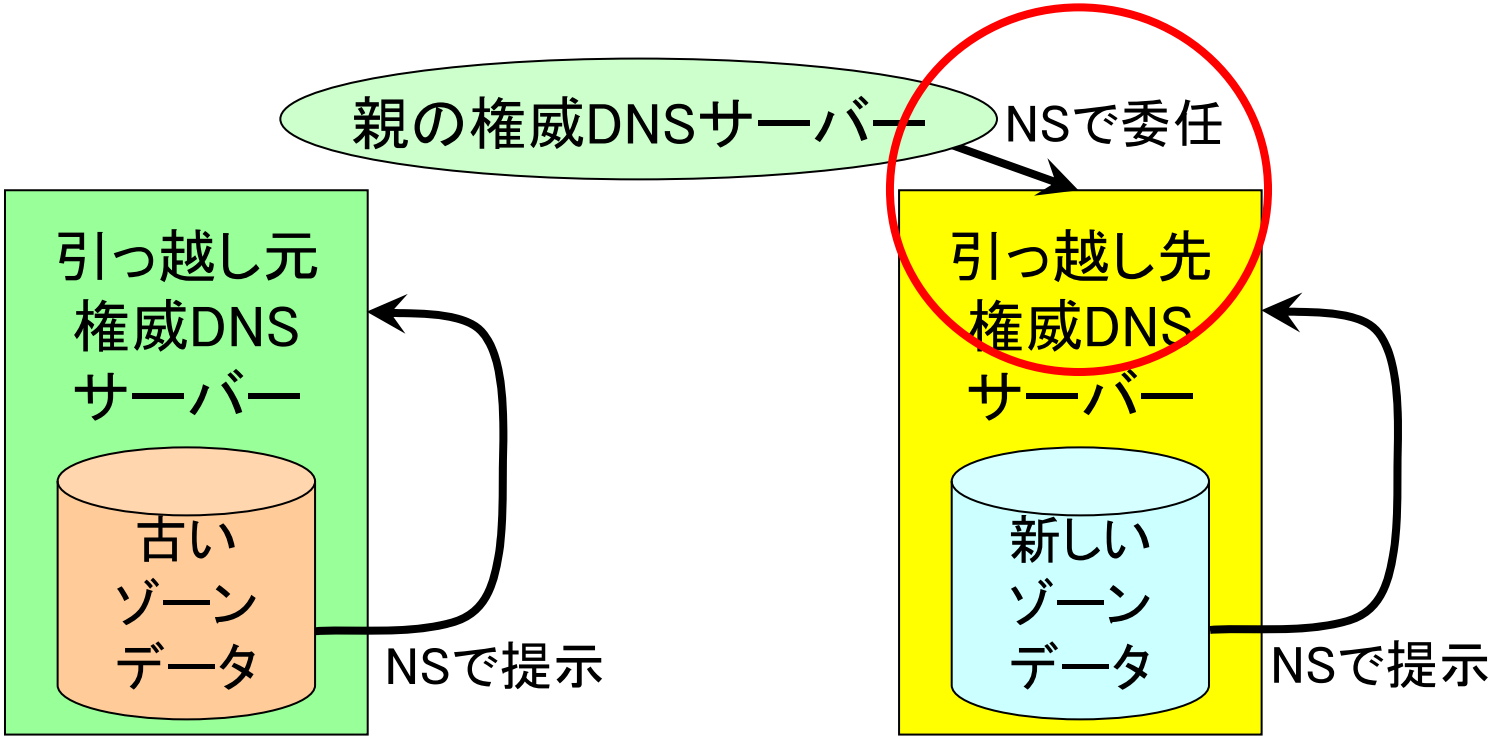


この方法のポイント

- **ゾーンデータの移行**を権威DNSサーバーの移行よりも**前に**実施する
  - 手順2の完了後、インターネット上のキャッシュDNSサーバー群には**新しいゾーンデータのみ**が提供されるようになる
- そのため、古いゾーンデータは各々のTTL値で指定されていた時間の経過後、**確実に消滅**する
- もし古いゾーンデータが消滅しなかった場合、当該キャッシュDNSサーバーの**動作不良であると断定**できる

浸透問題が起こりうる引っ越し方法

- 親のNSの切り替えだけを実施し、**引っ越し元の権威DNSサーバーの古いゾーンデータはそのまま**
- 実際の引っ越し（プロバイダの変更）でよくある形



この方法の何がいけないのか？

- 引っ越し元権威DNSサーバー（子）のNSが**既にキャッシュされている場合**に問題となる
  - 通常の名前検索において必ずキャッシュされる
- NSのTTLの満了よりも**前に**wwwなどのAのTTLが満了した場合、**NSで指定された引っ越し元権威DNSサーバー**にAを検索しにいく（これ自体は正しい動作）
- **古いA**が応答されキャッシュされる（**浸透問題その1**）
- その応答のauthority sectionには「私が確かに権威を持っています」という情報（NS）が入っており、実装によっては**キャッシュされているNSのTTL値がリセットされてしまう**（**浸透問題その2** ）

これが問題！

図解：これが浸透問題の正体！

1. 最初のキャッシュの状態がこうだったとする

```
www.example.jp. 10 IN A 192.0.2.1
example.jp. 100 IN NS ns-old.example.jp.
```

2. 10秒後に古いAレコードがキャッシュから消滅、90秒経過する前（例えば2秒後）に
ユーザーからの求めに応じ、www.example.jpを**ns-old.example.jp**に問い合わせ

```
（消滅）
example.jp. 90 IN NS ns-old.example.jp.
```

3. ns-old.example.jpからwww.example.jpの
古いIPアドレスと古いNSレコードを受け取る

```
（消滅）
example.jp. 88 IN NS ns-old.example.jp.
```

```
www.example.jp. 100 IN A 192.0.2.1
example.jp. 600 IN NS ns-old.example.jp.
```

4. **古いIPアドレスがキャッシュ**され、NSレコードのTTLが**リセットされる**（巻き戻る）

```
www.example.jp. 100 IN A 192.0.2.1
example.jp. 600 IN NS ns-old.example.jp.
```

これが**浸透問題の正体！**

図解：これが浸透問題の正体！

1. 最初のキャッシュの状態がこうだったとする

www.example.jp. 10 IN A 192.0.2.1
example.jp. 100 IN NS ns-old.example.jp.

2. 10秒後に古いAレコードがキャッシュから消滅、90秒経過する前（例えば2秒後）に
ユーザーからの求めに応じ、www.example.jpを**ns-old.example.jp**に問い合わせ

（消滅）
example.jp. 90 IN NS ns-old.example.jp.

＜重要なポイント＞
正しい方法では
ここで**新しい情報**を受け取るので…

3. ns-old.example.jpからwww.example.jpの
~~古い~~**IPアドレス**と~~古い~~**NSレコード**を受け取る

（消滅）
example.jp. 88 IN NS ns-old.example.jp.

www.example.jp. 100 IN A **192.0.2.100**
example.jp. 600 IN NS ns-**new**.example.jp.

新しい
4. ~~古い~~**IPアドレスがキャッシュ**され、NSレコードのTTLが~~**リセットされる**（巻き戻る）~~

新しいものに切り替わる

www.example.jp. 100 IN A **192.0.2.100**
example.jp. 600 IN NS ns-**new**.example.jp.

浸透問題は**発生しない！**

この動作はバグなのか？

- バグとは言い切れない
  - DNSプロトコルに違反しているわけではない
  - 正しい方法ではそもそも問題は発生しない
- 既にキャッシュされているデータと同じ信頼度のデータが来た場合にキャッシュDNSサーバーがどのようにふるまうかは、**DNSプロトコルでは決められていない**

つまり、浸透問題とは…

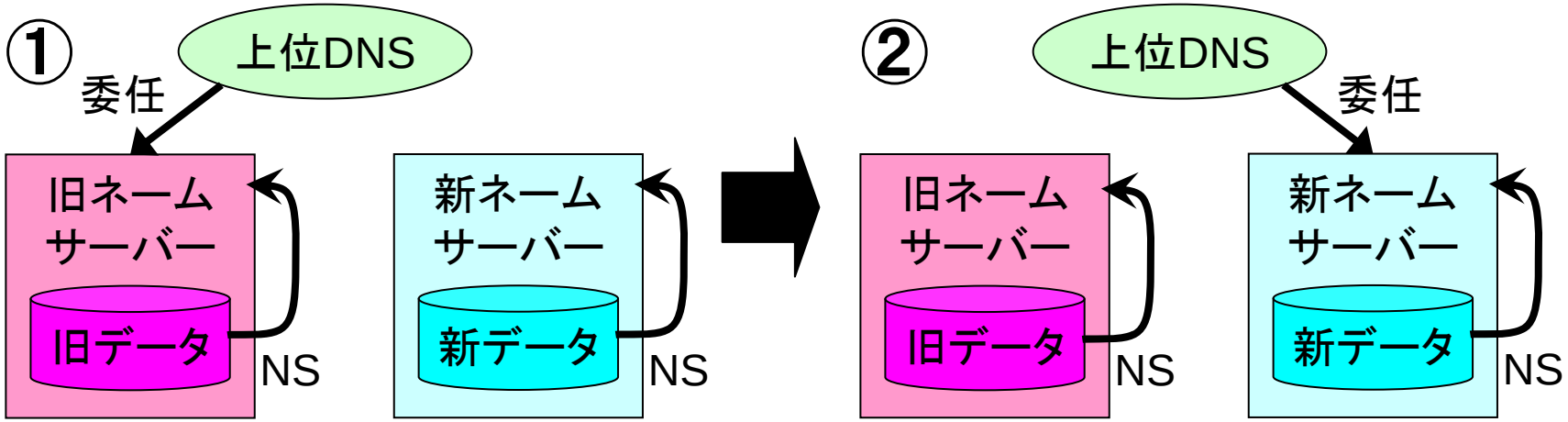
- 「動作が決められていないため実装依存である」ことと「正しい方法で引っ越しをしていない」ことの双方に起因する、**複合問題**である
- 正しい方法で引っ越しをすれば浸透問題は発生しない
- 実装によりNSのTTLリセットを回避することで、浸透問題の発生リスクを低減可能
  - では、どんな実装で浸透問題が発生するのか？

現時点における調査結果

- BIND 9
  - BIND 9.2.3で修正（TTLリセットを回避）された
    ⇒ BIND 9で浸透問題を起こすのは**9.2.2**まで
- Unbound
  - Unboundでは浸透問題は起こらない
- Google Public DNS
  - 浸透問題が起こる場合があるという指摘あり
  - 要詳細調査

こんな実験環境を用意してみました

- 旧・新のゾーンデータを変更せず、
  上位側の**委任先IPアドレスのみ**変更
  - 対象ドメイン名　　www.ex.t.dnslab.jp (TTL 5秒)
  - IPアドレス　10.111.111.111 ⇒ 10.222.222.222
  - TTL:　　上位 15秒　　下位 10秒 (wwwを除く)



ということで、
論よりRun ☺

その前に…

論よりRunの見どころ

- digコマンドを使い対象キャッシュサーバで
  www.ex.t.dnslab.jp を検索
- 上位NSのA RRを実験開始直後に切り替え
- IPアドレスはどのタイミングで切り替わるのか

  旧　10.111.111.111

  ↓

  新　10.222.222.222

  ====== 数字 ======：　経過時間(秒)

（BIND 9.8.1-P1 & 9.2.2による浸透問題のデモ）

まだたくさんある古いBIND

- BINDではずいぶん前に修正されているのに、未だに「DNSが浸透しない」などと騒がれるのはなぜか？
- インターネットにおけるBIND 9.2系までのシェア
  - BIND 9全体の**約33%**（JPRS調べ）
  - 例えばRed Hat Enterprise Linux 3系のやや古いものを使いつづけているとか…
  - 多くのメーカー製OSの場合、BINDのセキュリティホールは独自に修正されるが、**バージョンアップは行われない**

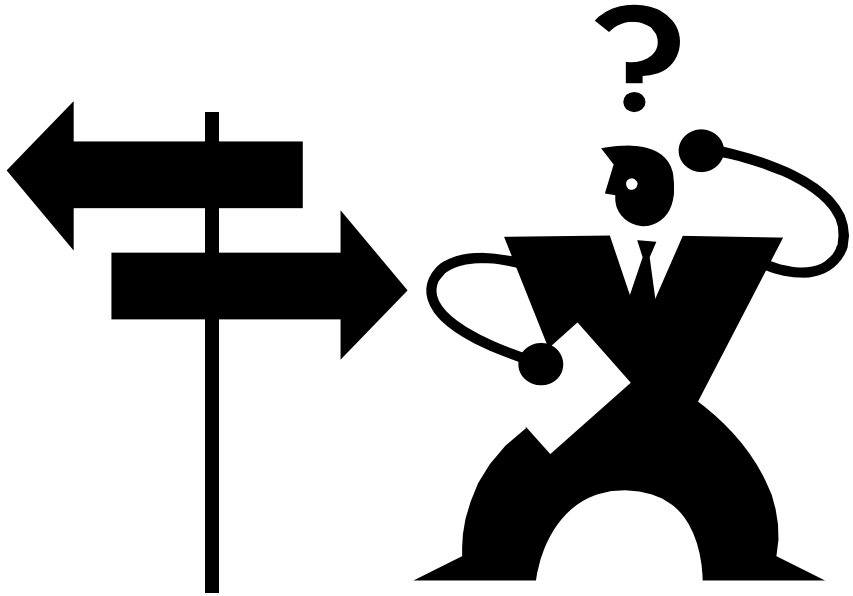
まとめ

- 浸透問題ではなく「消えない問題」
  - 新しいデータではなく**古いデータ**に着目すべし
- 浸透問題は複合問題
  - **正しい方法で引っ越しをしていない**
  - **古いBINDを使い続けている**組織が多い
- 浸透問題の解決は簡単ではない
  - 正しい方法で引っ越しをするのは難しい
    - 技術的に難しいのではなく、**運用・しくみ的**に難しい
  - 古いBINDがなくならない限り、リスクはなくならない

おすすめ

- 古いBIND 9は捨てましょう
  - 動いているだけで**有害**です
  - あなたの周りに古いサーバーはありませんか？
  - 古いLinuxディストリビューションを使い続けていませんか？
- 古い権威DNSサーバーのデータは**有害**です
  - インターネット全体に迷惑がかかります
- 可能であれば、正しい引っ越しをしましょう
  - 浸透問題は**避けられる問題**です
- DNSに対する正しい知識を**浸透**させましょう
  - **正しい知識の浸透**が浸透問題の発生を減らします

ありがとうございました！



＜会場のみなさまへ＞

- Q&A、コメント
- 浸透問題の正体を世の中に浸透させるためには？
- 正しい引っ越し方法を世の中に浸透させるためには？