DNSキャッシュサーバ設計と運用のノウハウ

東　大亮

DNS Summer Days 2014

2014 / 6 / 26

はじめに

- DNSが正しく動作しないと、インターネットのサービス利用に多大な影響が出る

- キャッシュサーバは、DNSの重要な要素の一つだが、安定動作のためのノウハウがあり、個別に語られることはあるが、まとまったものはあまり無い（と思う）

DNSキャッシュサーバを動かすだけなら簡単

- BIND9もUnboundも、設定ファイル無しでキャッシュサーバとして動いてしまう

```
# named -c /dev/null

# unbound -c /dev/null
```

“-c”は設定ファイルの指定

BIND9やUnboundは、（その版がリリースされた時点での）ルートヒントも内蔵しているので、設定情報としてルートヒントを与える必要もない

上記設定では、localnets; localhost (BIND9)・localhost (Unbound)からの再帰クエリを受け付けるDNSキャッシュサーバとして動作

動かすだけなら簡単…でも、

- デフォルト設定は、ハードウェアの性能を最大限に発揮できるとは限らないため、ある程度の規模以上の用途ではチューニングが必要

- トラブルを防ぐための、ネットワーク環境も含めた正しい設定が必要

- それでも発生し得るトラブルに対する備えもあったほうがよい

今日のお話

- トラブル無く（ちゃんと名前解決できる）、安定したDNSキャッシュサーバを運用するために知られているノウハウをいくつか紹介
- パフォーマンスチューニング
- トラブルを避けるための設計と運用
- （それでも起こってしまう）トラブルに備える
- セキュリティについて

注意！

- 本発表では、具体的なDNSキャッシュサーバの設定や構成の例を多く紹介しますが、全ての環境に適する設定とは限りません（きわどいのもあります）。
- この発表の設定例を実環境に投入する場合は、事前に十分な検証を行ってください。
- 本発表ではキャッシュサーバの実装として利用者が多いBIND 9とUnboundの設定例を含みます。各実装のバージョンは以下の通り。
  - BIND 9.9.5-P1
  - Unbound 1.4.22
- 本発表はDNSキャッシュサーバ専用で動作しているDNSサーバを前提としています。**権威ネームサーバの話はしません。**

パフォーマンス
チューニング

パフォーマンスチューニングの必要性

- 多くのDNSキャッシュサーバ実装のデフォルト設定は、システムの資源を浪費しないように**リミッター**がかかっている

- **ハードウェアの性能を最大限に発揮できる設定ではない**ため、チューニング不足だと以下の現象が起こる：

  - **CPU使用率が高くないのに、クライアントからの再帰検索要求をドロップしてしまう**

  - **秒間あたりの再帰検索要求(QPS)が低いのに、なぜかCPU使用率が高い**

  - **検索速度が遅い（キャッシュヒット率が上がらない）**

- トラフィックが小さい(〜数百QPS)ならばチューニングは不要だが、大きくなってくるとチューニングは必要

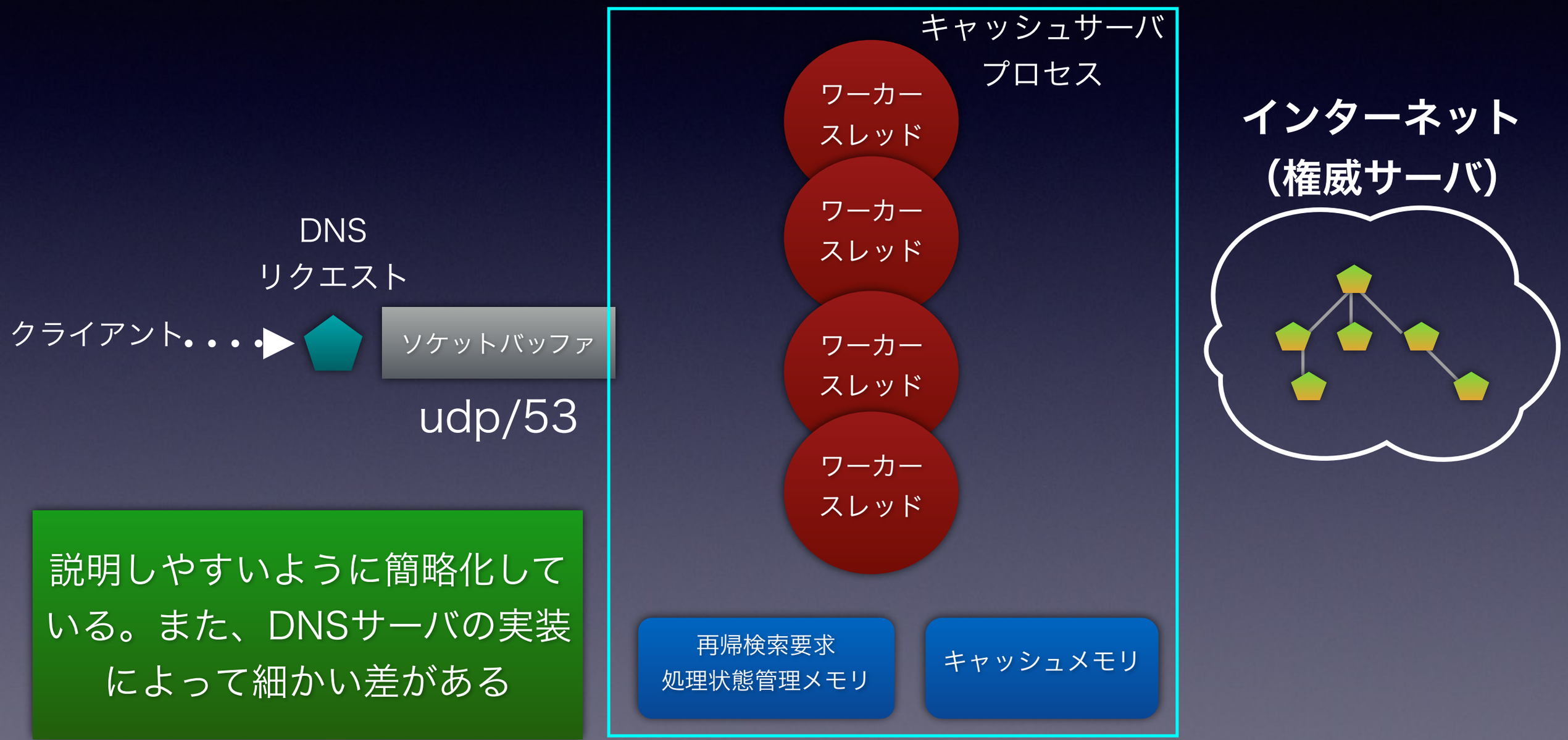
「DNSキャッシュサーバチューニングの勘所」

- キャッシュサーバのチューニングは、これだけのテーマでお話しても1時間以上かかるネタです
- というわけで別冊にしました
  - http://www.slideshare.net/hdais/dns-32071366
  - 「DNSキャッシュサーバ　チューニングの勘所」で検索すればslideshareで出てきます
- 以下は**ハマりやすいポイント**をご紹介します

チューニング
これだけはやっておこう(1)

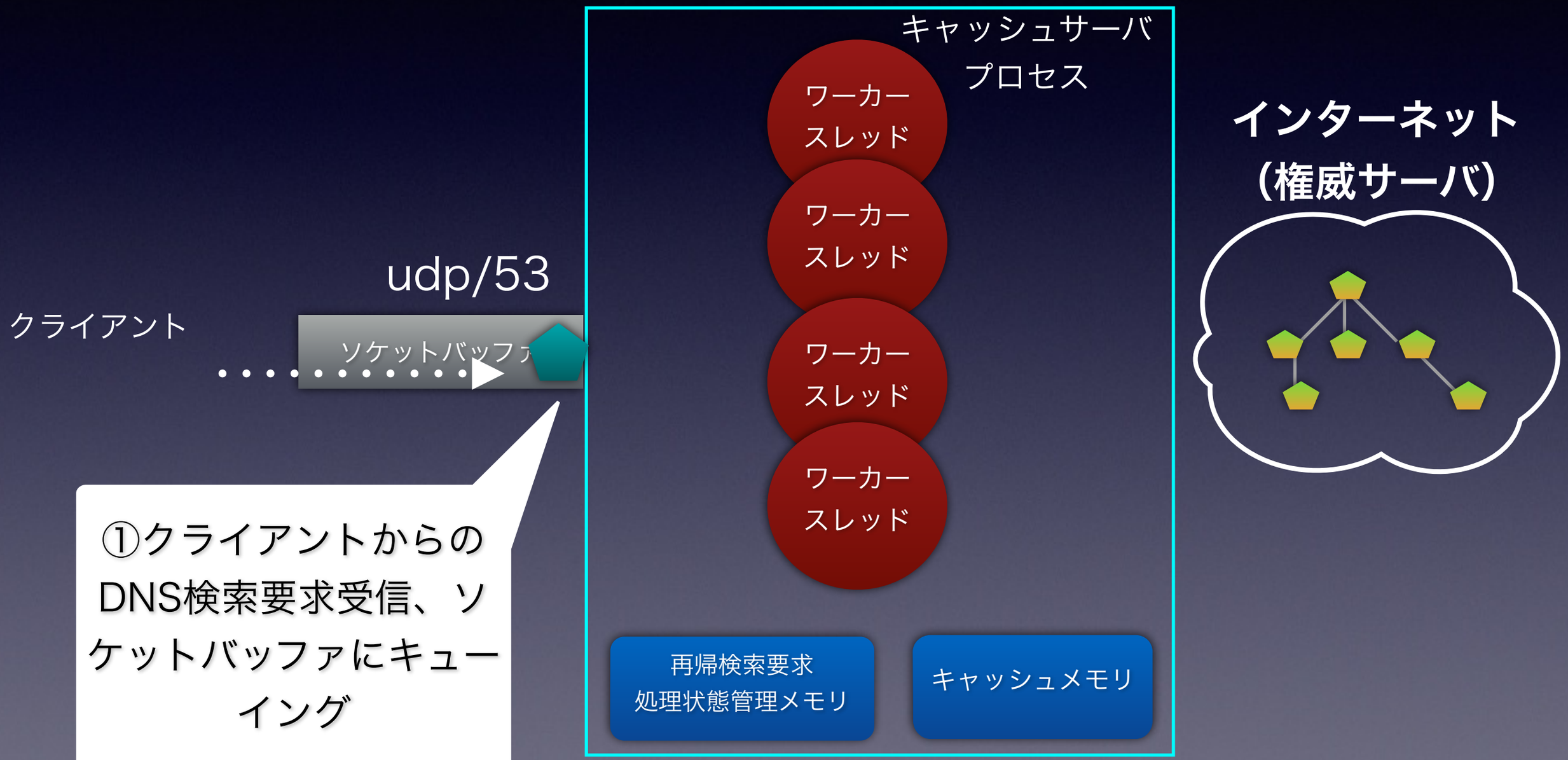
- クライアントから受信した、未解決（権威サーバに問合せ処理中）の再帰検索要求の処理状態を管理・保持する領域のサイズの引き上げ
  - BIND9のデフォルト値 recursive-clients 1000
  - Unboundのデフォルト値 num-queries-per-threads 512 or 1024
- 数千QPS以上のDNSキャッシュサーバでは小さすぎるため、適切な値に**引き上げる**必要がある

DNSキャッシュサーバ
内部構造

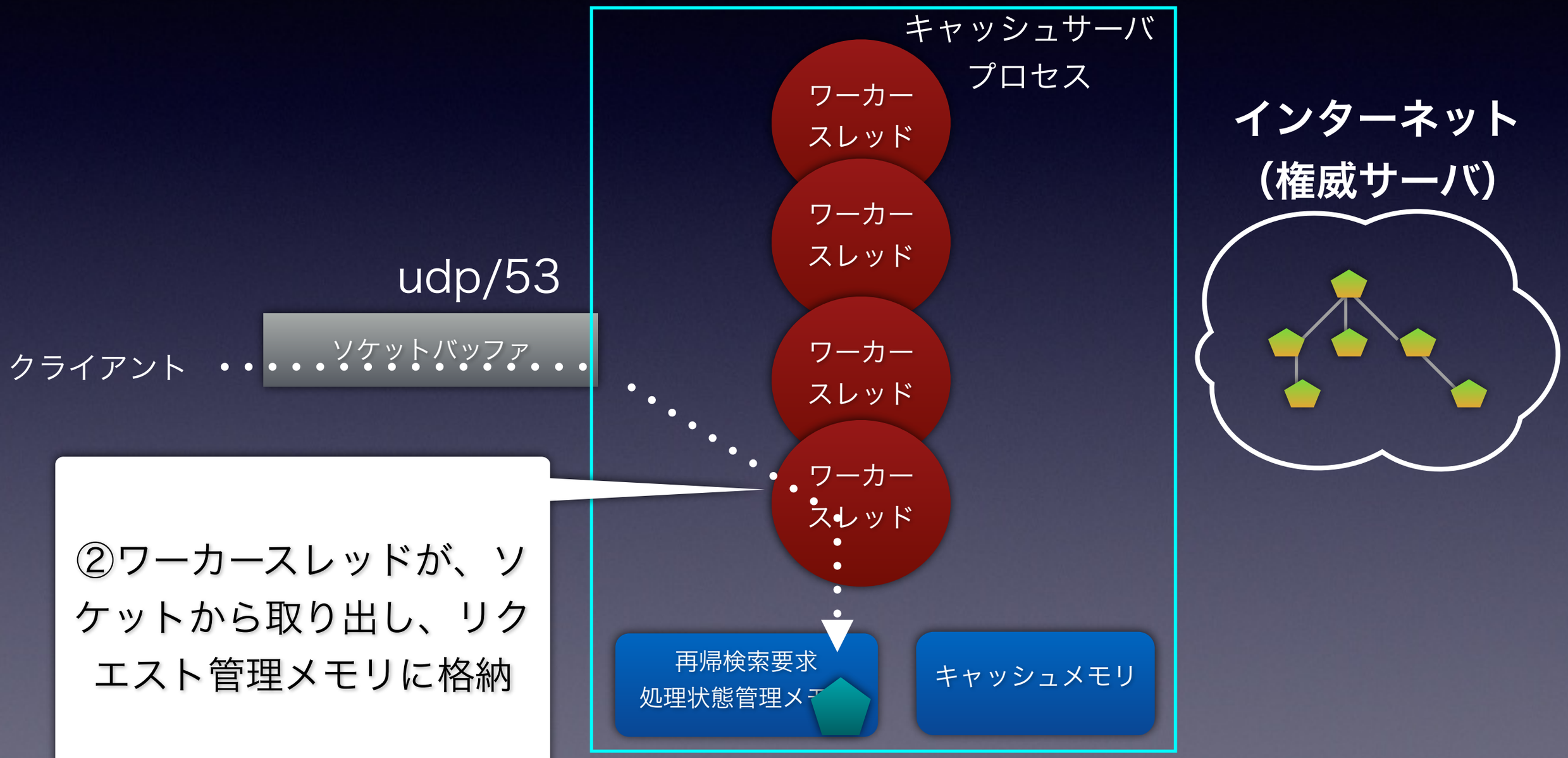


説明しやすいように簡略化している。また、DNSサーバの実装によって細かい差がある

DNS再帰検索処理(1)

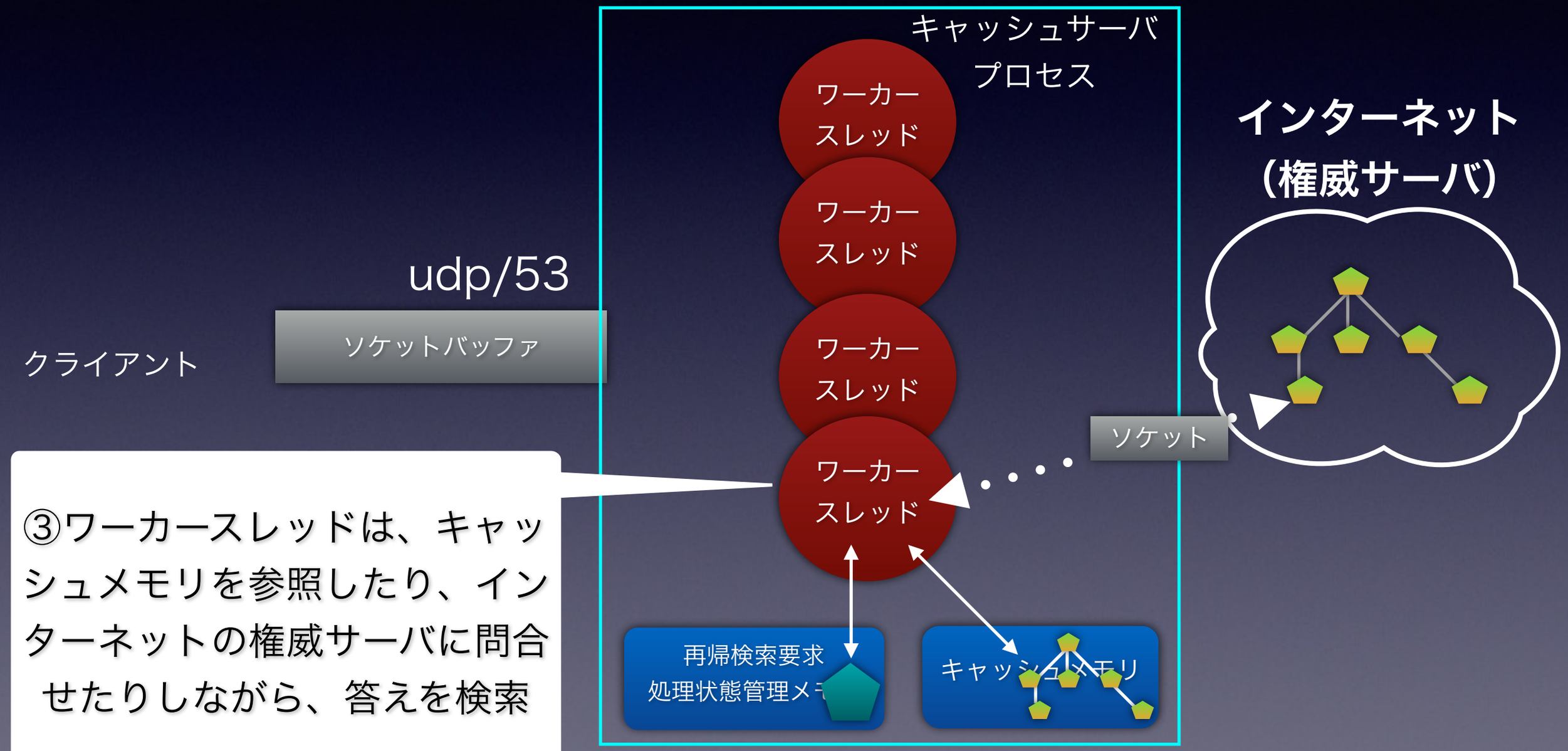

キャッシュサーバ
プロセス
ワーカー
スレッド
ワーカー
スレッド
ワーカー
スレッド
ワーカー
スレッド
インターネット
（権威サーバ）
udp/53
クライアント
ソケットバッファ
①クライアントからのDNS検索要求受信、ソケットバッファにキューイング
再帰検索要求
処理状態管理メモリ
キャッシュメモリ

DNS再帰検索処理(2)



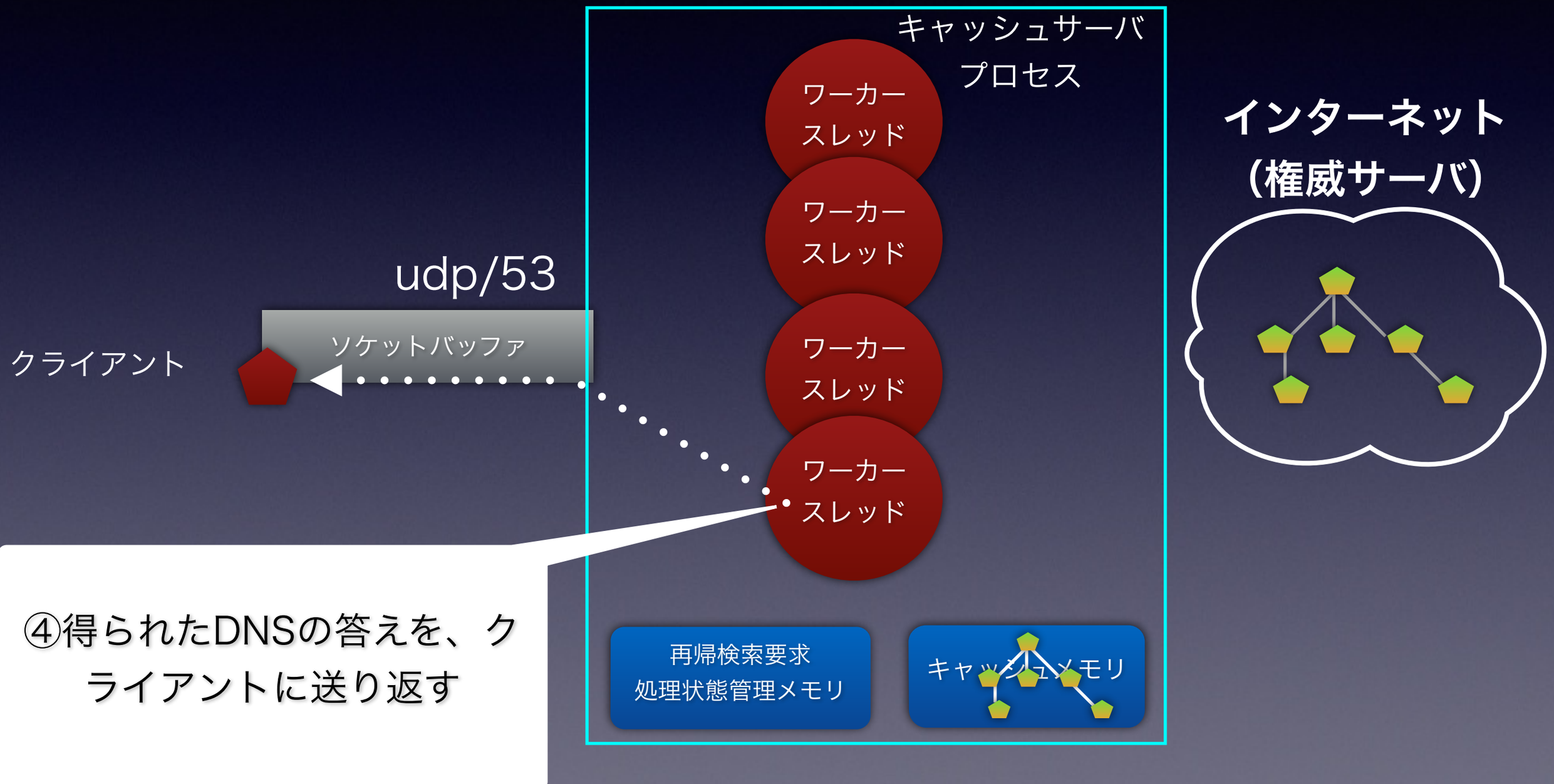
②ワーカースレッドが、ソケットから取り出し、リクエスト管理メモリに格納

DNS再帰検索処理(3)



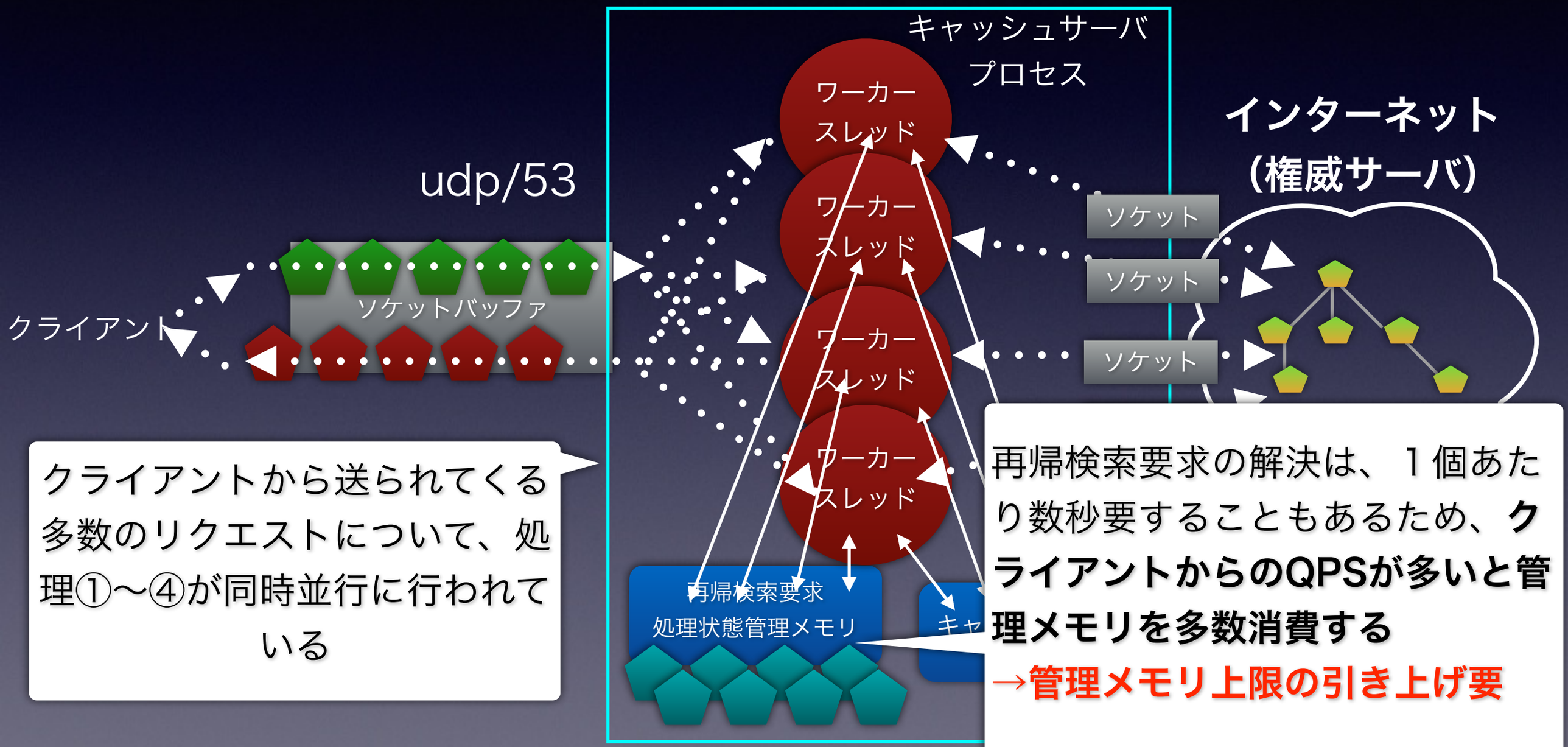
③ワーカースレッドは、キャッシュメモリを参照したり、インターネットの権威サーバに問合せたりしながら、答えを検索

DNS再帰検索処理(4)



④得られたDNSの答えを、クライアントに送り返す

高負荷なDNSキャッシュサーバの状態と、要求処理状態管理メモリ量の引き上げ



クライアントから送られてくる多数のリクエストについて、処理①～④が同時並行に行われている

再帰検索要求の解決は、１個あたり数秒要することもあるため、**クライアントからのQPSが多いと管理メモリを多数消費する**

→管理メモリ上限の引き上げ要

チューニング
これだけはやっておこう(2)

- **キャッシュメモリのサイズ**
  - **BIND9**のデフォルト値　max-cache-size **制限無し**(キャッシュされたRRSetのTTLが切れるまで保持)
    - システムの**メモリを食い尽くしてしまう**おそれがあるため、**制限をかける**
  - **Unbound**のデフォルト値 rrset-cache-size: 4MB / msg-cache-size: 4MB
    - 4MB / 4MBは多数のクライアントを収容するキャッシュサーバとしては**小さすぎる**ため、引き上げる

トラブルを避ける設計と運用

DNSキャッシュサーバの特徴

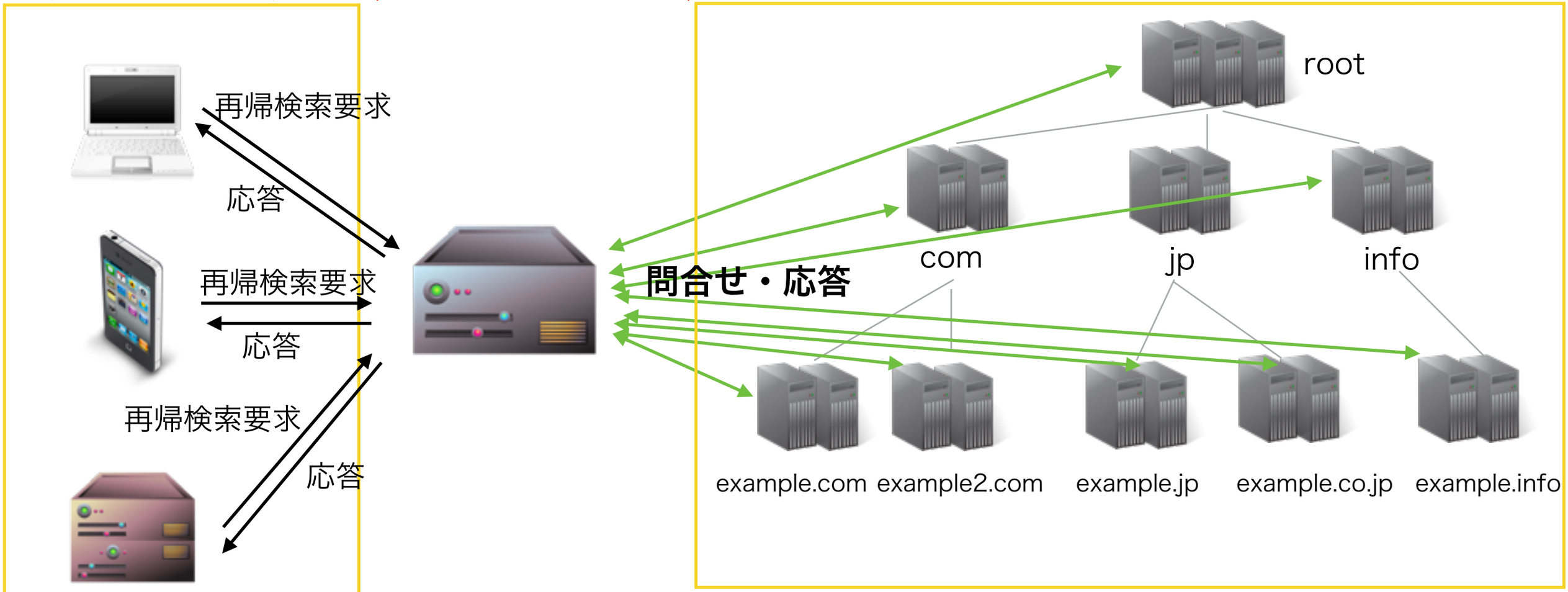
全体外観

クライアント側と権威ネームサーバ側の通信に大きく分かれる

クライアント
(スタブリゾルバ)

キャッシュサーバ
(フルサービスリゾルバ)

権威ネームサーバ
(インターネット上に分散配置)

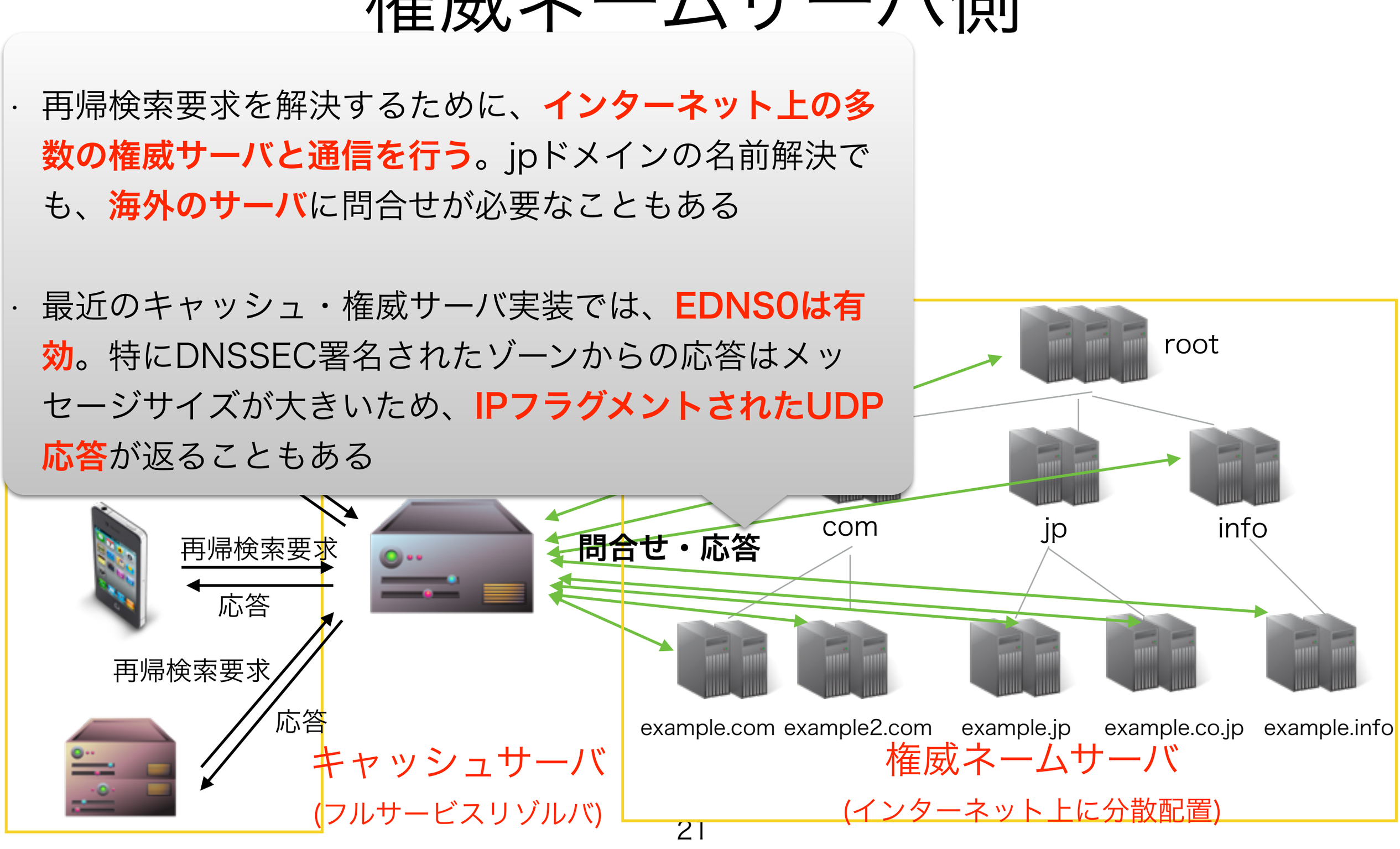


DNSキャッシュサーバの特徴
クライアント側

- クライアント（スタブリゾルバ）からの再帰検索要求を受信、検索結果を応答
- クライアントがDNS名前検索が必要となるたびに要求がある
- ISP環境のキャッシュサーバでは、多数のクライアントを収容するため、**秒間あたりの要求数 (QPS)が大きい**
- **EDNS0オフ**のクライアントが多い (**UDPメッセージは512バイトまで**)

DNSキャッシュサーバの特徴
権威ネームサーバ側

- 再帰検索要求を解決するために、**インターネット上の多数の権威サーバと通信を行う**。jpドメインの名前解決でも、**海外のサーバ**に問合せが必要なこともある
- 最近のキャッシュ・権威サーバ実装では、**EDNS0は有効**。特にDNSSEC署名されたゾーンからの応答はメッセージサイズが大きいため、**IPフラグメントされたUDP応答**が返ることもある



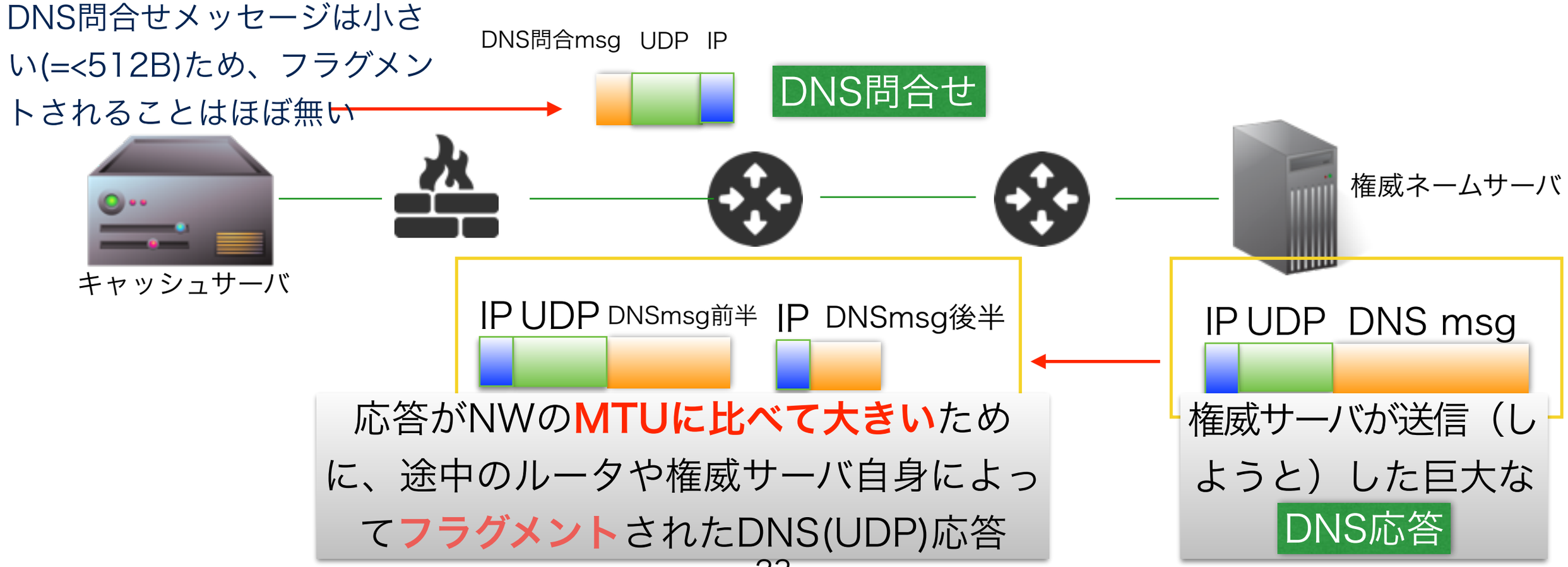
DNSキャッシュサーバにおけるトラブル

- トラブルは無数に起こりえるが、ありがちなトラブルと、それを防止・回避する設定を紹介
- IPフラグメントが届かない問題の回避
- TCPクエリに対応しないクライアントの問題の回避

IPフラグメントが届かない問題(1)

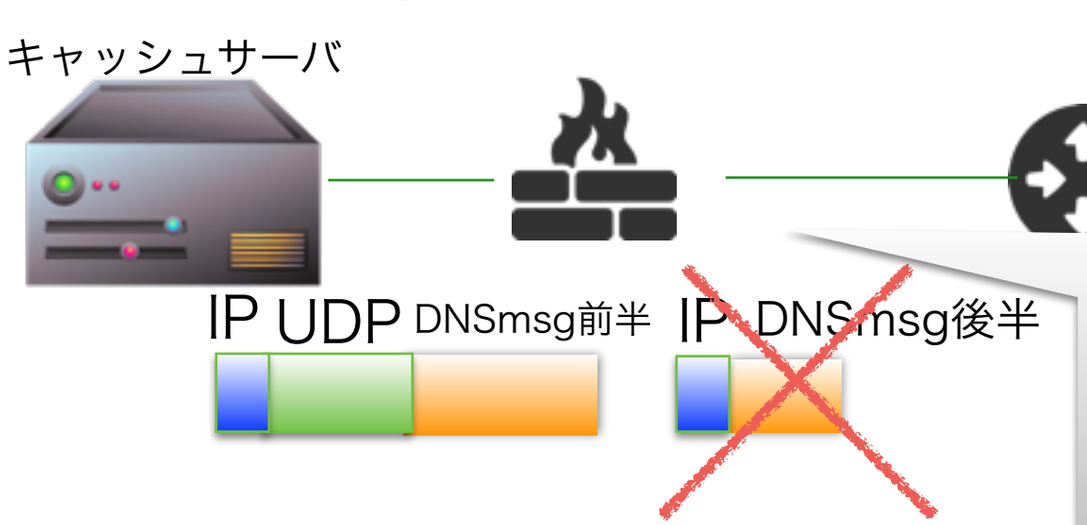
- 権威サーバ→キャッシュサーバ方向の**DNS応答が巨大**な場合、**UDP応答がIPフラグメント化されてキャッシュサーバへ送信される**ことがある

- 検索対象ドメインがDNSSEC署名されていた場合に多い



IPフラグメントが届かない問題(2)

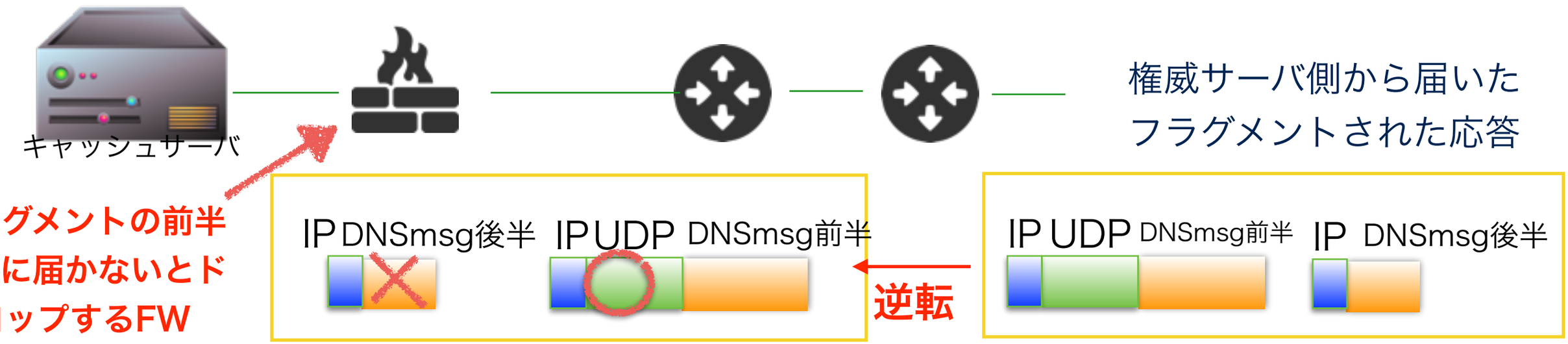
- 途中のネットワーク経路上の問題により、**IPフラグメントが疎通せず、キャッシュサーバが応答を受け取れない**ことがある→**対象ドメイン名の解決が不可**(SERVFAIL)になる
- **「おたくのキャッシュで特定のドメイン名が引けないんだけど（他では引ける)」系トラブルの原因！**
  - フラグメントされるような**巨大なDNS応答の時だけ問題が起こるため、厄介**
  - **以前から指摘されていますが、今でもDNS技術系ML等でも現役の頻出Q&Aです（私もハマりました)**



- **ありがちなパターン**：経路上の**ファイアーウォール**や**ロードバランサ**の設定ミスやバグでフラグメントを正しく転送できない

IPフラグメントが届かない問題(3)

- 通常は問題ないよう見えても、**FWやLBの設定不良やバグ**により特定の条件でフラグメントが疎通できない場合がある
- **FWの不具合事例**（私の自宅のBBルータのNAT機能と、某FW製品が該当…）
  - フラグメントの**先頭が先に届かないと、2番目以降のフラグメントを通さないFW**が存在する
  - インターネットの経路によっては**フラグメントが高確率で順序が逆転して届くことがある**ため、そのようなFWがあると**フラグメントが疎通しない**



IPフラグメントが届かない問題(4)

- FWだけでなく**ルータやL3SWのACL**においても、フラグメントを意図通りに通さないものも存在する
- 以下は53/udp宛のパケットを通すACLであるが、**IPフラグメント（特に２番目以降のフラグメント）が許可なのか拒否なのかは、ベンダによって異なる**

```
access-list 100 permit udp any any eq 53
```

IPフラグメントが届かない問題(5)

- **レイヤ4以上のフィールド**を見てアクセス制限やフォワードしている部分については、**IPフラグメントの扱いを確認**しましょう

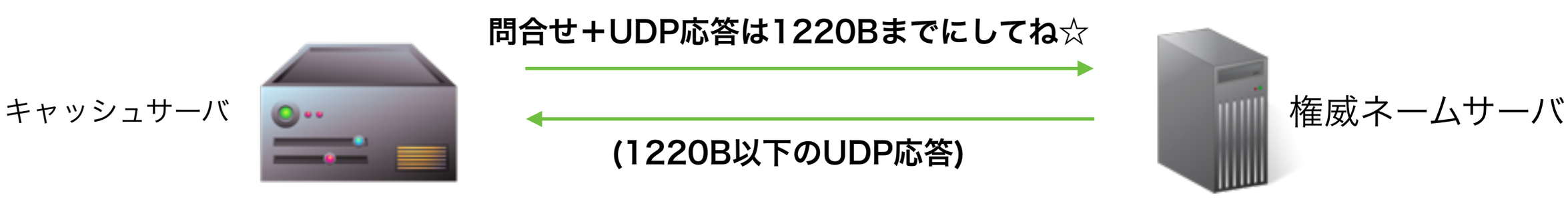
- **LBやFWの選定・設定・検証**においては、**IPフラグメントの扱いに細心の注意**を払いましょう

- **LBやFWのベンダ様各位**におかれましては、**IPフラグメント周りのバグには特にご注意**頂きますようお願いいたします（実装が大変なのは、分かりますけど）

- 些細な問題のようですが、**ハマるとかなりきついです**

IPフラグメントが届かない問題(6)

- **フラグメント問題を解消できない場合の回避策**
  - キャッシュサーバが権威ネームサーバに問合せる時に、EDNS0で**UDP応答サイズの最大値を指定可能**(BIND9/Unboundともにデフォルト4096B)。
  - **この値をフラグメントされないサイズまで小さくする**ことで、フラグメントを回避できる
    - DNS応答が指定サイズに収まらない場合は、TCPで再問合せ(TCPフォールバック)



IPフラグメントが届かない問題(7)

応答サイズ指定方法

BIND9

権威サーバを特定(192.0.2.1)して応答サイズ指定する場合

```
server 192.0.2.1 {
    edns-udp-size 1220;
};
```

全権威サーバに対して共通で指定する時はoptions節で記載

```
options {
    edns-udp-size 1220;
};
```

Unbound

Unboundは、**全権威サーバ共通指定のみ可**

```
edns-buffer-size: 1220;
```

UDP応答が1220バイトに収まらずTCP問合せが行われると負荷が高いので、可能なら権威サーバ指定が良い

Unbound/BIND9ともに、未指定時のデフォルトは4096バイト

IPフラグメントが届かない問題(8)

DNSキャッシュサーバのFWが、フラグメントを通すかどうかの簡易試験ツール

DNS-OARCのサービスが便利

https://www.dns-oarc.net/oarc/services/replysizetest

ただし、逆転したフラグメントが通らない等の特定条件の問題は見つからないこともあります

IPフラグメントが届かない問題(9)

キャッシュサーバから特定の権威サーバへの通信が、フラグメント問題に該当しているか調べる方法

キャッシュサーバ上から、digで問題の権威サーバにDNS問合せを送信してみる。
このとき DO=1 (DNSSEC OK)かつEDNS0バッファサイズを4096に指定
(BIND9 / Unboundのデフォルト)

```
① dig @192.0.2.1 www.example.com A +norec +dnssec +bufsize=4096
```

次にバッファサイズを小さくして問合せてみる

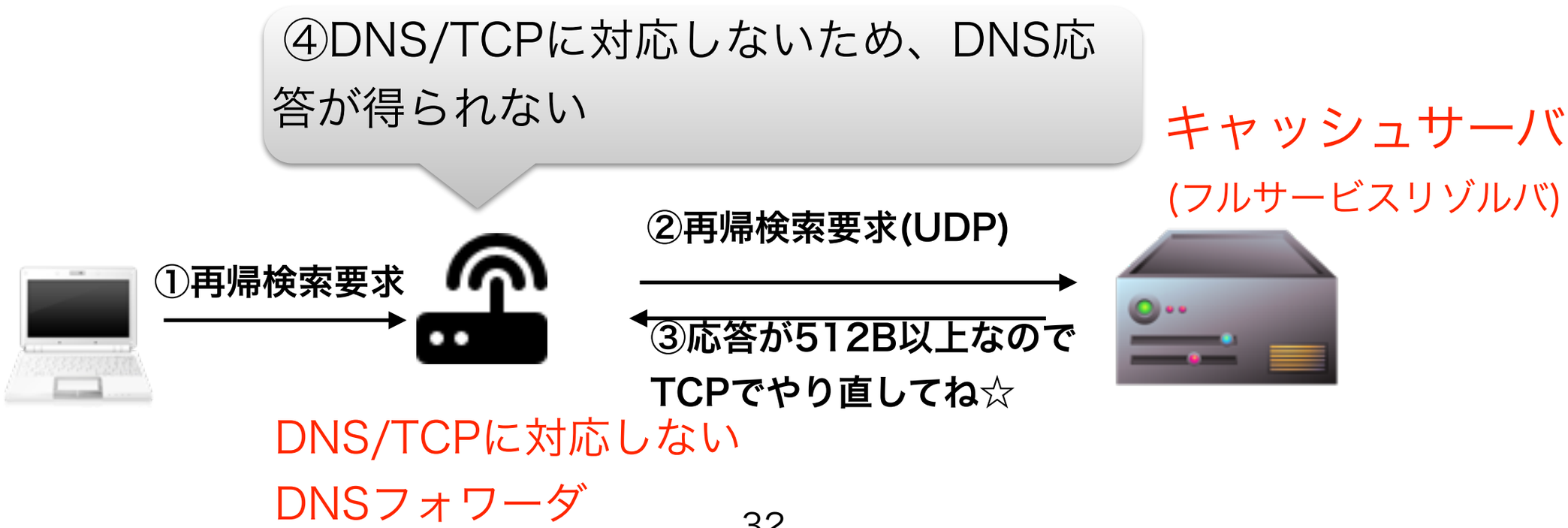
```
② dig @192.0.2.1 www.example.com A +norec +dnssec +bufsize=1220
```

①で応答が得られず、②で応答(TCPフォールバック後)が得られた場合、フラグメント問題に該当している可能性が高い

TCPに対応しないクライアントの対応(1)

- クライアント(スタブリゾルバ)⇔キャッシュサーバ間は、**EDNS0無効**のことが多く、**DNS応答が512Bを超える時はTCPが使われる**

- **TCPに対応せず、512Bを超える応答が扱えないクライアントが存在**し、そのようなクライアントでは**名前解決できないドメインが存在**する
  - 家庭用ブロードバンドルータ内蔵のDNSフォワーダに多い



TCPに対応しないクライアントの対応(2)

- pixiv.netの事例

  pixivでは**DNSラウンドロビン**を限界まで使ってみたことがあるという。ぶら下げる**Webサーバの数が一定数を超えると「DNSレコードがUDPパケットに収まらなくて、ユーザーの環境によっては“画像が見えない”ということが起こった」**という。**DNSでは応答が512バイトを超えると、TCPにフォールバックしようとするが、家庭用ルータにはこの仕組みがうまく扱えないものもあるからだ。**

引用(強調は東)：http://www.atmarkit.co.jp/news/201007/21/pixiv.html

2010年の記事ですが、現在でも同じ原因で、他のドメイン名が「引けない」という騒ぎが時々あります。

TCPクエリに対応しないクライアントの対応(3) minimal-responses

- DNS応答のauthority section/additional sectionを(プロトコル上許される範囲で)省略する機能
- DNS応答サイズが削減され、TCPが必要になる可能性が減る
    - BIND9　　minimal-responses yes;
    - Unbound　minimal-responses: yes
- 常にminimal-responses応答を行うDNSキャッシュサーバも多い
    - PowerDNS recursor、dnscache (djbdns)、Google Public DNS等

TCPクエリに対応しないクライアントの対応(4)
minimal-responsesの効果

- 応答からAuthority/Additional sectionが省略され、
  DNS応答サイズが小さくなっている

minimal-responses no

```
$ dig www.kernel.org

;; QUESTION SECTION:
;www.kernel.org.        IN  A

;; ANSWER SECTION:
www.kernel.org.     600 IN  CNAME  pub.all.kernel.org.
pub.all.kernel.org. 600 IN  A   198.145.20.140
pub.all.kernel.org. 600 IN  A   149.20.4.69
pub.all.kernel.org. 600 IN  A   199.204.44.194

;; AUTHORITY SECTION:
kernel.org.      86400  IN  NS  ns0.kernel.org.
kernel.org.      86400  IN  NS  ns1.kernel.org.
kernel.org.      86400  IN  NS  ns2.kernel.org.

;; ADDITIONAL SECTION:
ns0.kernel.org.     86400  IN  A   198.145.19.196
ns1.kernel.org.     86400  IN  A   149.20.20.144
ns1.kernel.org.     86400  IN  AAAA   2001:4f8:8:10::1:1
ns2.kernel.org.     86400  IN  A   149.20.4.80
ns2.kernel.org.     86400  IN  AAAA   2001:4f8:1:10::1:1

;; MSG SIZE  rcvd: 260
```

minimal-responses yes

```
$ dig www.kernel.org

;; QUESTION SECTION:
;www.kernel.org.        IN  A

;; ANSWER SECTION:
www.kernel.org.     600 IN  CNAME  pub.all.kernel.org.
pub.all.kernel.org. 600 IN  A   199.204.44.194
pub.all.kernel.org. 600 IN  A   198.145.20.140
pub.all.kernel.org. 600 IN  A   149.20.4.69

;; MSG SIZE  rcvd: 102
```

512バイト問題を100%回避できるとは限らないが、効果は高い

それでも起こる
トラブルへの備え

トラブルに備える

- DNSキャッシュサーバの監視
- トラフィックログの取得

DNSキャッシュサーバの監視(1)

- **基本的な監視**
  - CPU、メモリ使用量、ロードアベレージ、ネットワークトラフィック量
- **キャッシュサーバの正常性の監視**
  - 各種統計情報の取得
  - ドメイン名が正しく引けるか？

DNSキャッシュサーバの監視(2)

統計情報

- 様々な**統計情報**が有るが、モニタしておくとよい項目がある

- 異常時にすぐに状況が確認できるように、MRTG的なもので**グラフ化**すると良い

- **閾値監視**（一定値範囲から外れたらアラーム）もするとよいが、**負荷状況によって正常値が上下するので難しいかも**

DNSキャッシュサーバの監視(3)

統計情報

	BIND9 XML stats (isc/bind/statistics/server/nsstat/)	Unbound unbound-control stats_noreset
検索要求数	ΔRequestv4+ΔRequestv6	Δtotal.num.queries
キャッシュヒット率	(ΔRequestv4+ΔRequestv6 - ΔQryRecursion) ÷ (ΔRequestv4+ΔRequestv6)	Δtotal.num.cachehits ÷ Δtotal.num.queries
成功した検索要求 **(QNAME&QTYPEが見つかった)**	ΔQrySuccess	Δnum.answer.rcode.NOERROR - Δnum.answer.rcode.nodata
NODATA **(QNAME存在するがQTYPE無)**	ΔQryNxrrset	Δnum.answer.rcode.nodata
NXDOMAIN **(QNAME無し)**	ΔQryNXDOMAIN	Δnum.answer.rcode.NXDOMAIN
その他エラー	ΔQrySERVFAIL, ΔQryFORMERR	Δnum.answer.rcode.SERVFAIL, Δnum.answer.rcode.FORMERR
検索要求管理メモリの長さ	rndc status コマンドの出力の “recursive clients”	total.requestlist.current.all

「Δ」は前回サンプリング時からの増分値を表す。キャッシュヒット率・検索要求管理メモリの長さ以外については、値をサンプリング時間で割って単位時間あたりの値をモニタ対象にしてもよい

DNSキャッシュサーバの監視(4)
統計情報の激変時

	極端な増加時に考えられる事象	極端な減少時に考えられる事象
検索要求数	キャッシュサーバへの大量トラフィック	NW障害等でクライアントがキャッシュサーバに到達不能／大規模イベント（W杯等）
キャッシュヒット率		キャッシュサーバに対するDoS／NW障害等でキャッシュサーバが権威サーバに到達不能／キャッシュポイズニング攻撃
NXDOMAIN	キャッシュサーバに対するDoS／キャッシュポイズニング攻撃	
その他エラー	SERVFAIL: NWの障害でキャッシュサーバが多数の権威サーバに到達できない／キャッシュへのDoS攻撃	
検索要求管理メモリの長さ	キャッシュサーバに対するDoS攻撃／キャッシュサーバが権威サーバに到達不能	

DNSキャッシュサーバの監視(5)
ドメイン名が正しく引けるか？

- キャッシュサーバの正常性監視として、当該キャッシュサーバで**複数のドメイン名が正常に引けるか常時監視**すると良い

- **監視対象として適当と思われるもの**
  - **Alexa Top 500の上位ドメイン** (google.com, facebook.com, youtube.com, …)
    - http://www.alexa.com/topsites
  - **TTLが短め(10秒程度)のドメイン名**(自前で作ってもよい)
    - キャッシュメモリからの応答ではなく、キャッシュサーバ⇔権威サーバ間の再帰検索による応答が可能であるかの監視

トラフィックログの取得(1)

- 障害調査のため、DNSキャッシュサーバがやりとりしているトラフィックログを取得することは有益だが、以下の問題がある
  - BIND9もUnboundも、クライアントからの検索要求ログの取得は可能だが、トラフィックが多いサーバでは**ログの出力自体が大きな負荷**になり現実的ではない
    - 上記を理由として、敢えてクエリログ機能を実装していないDNSサーバもある(NSD等)
  - キャッシュサーバ⇔権威サーバ間の問合せ・応答を出力することはデバッグレベルのログが必要なことが多く、通常運用状態では現実的ではない

トラフィックログの取得(2)

- トラフィックが多いDNSキャッシュサーバを収容するL3SW/L2SWは、**ポートミラーリング**が可能なものにしましょう
  - DNSサーバに影響を与えずにトラフィックログを取れます
  - キャッシュサーバ自体でtcpdump動かしてディスクに貯めるのも負荷になるので
- 可能であれば、**パケットキャプチャ用のマシンをあらかじめ用意**して、ミラーポートに接続しておくくらいでもいいです

キャプチャしたパケットの解析

- **Wireshark**
  - フラグメントしたDNSパケットでもリアセンブルして解析できるので便利。巨大なpcapファイルは取込めないことがあるので、一旦tcpdumpでパケットキャプチャした内容を、tcpdump -r -w で特定の通信のみフィルタしてWiresharkに入力すると良い
- **dnstop／dsc**
  - DNSパケットをキャプチャしながら統計を取ることができるツール
    - dsc http://dns.measurement-factory.com/tools/dsc/
    - dnstop http://dns.measurement-factory.com/tools/dnstop/

セキュリティについて

セキュリティについて

- DNSキャッシュサーバにおける一般的なセキュリティ関連対策について
  - アクセス制限（オープンリゾルバにしない）
  - キャッシュポイズニング対策
    - ポートランダマイズ
    - ポートランダマイズ以外の対策

アクセス制限の必要性

- インターネット上のどのクライアントからも再帰検索要求を受け付けるキャッシュサーバ（オープンリゾルバ）を構築してはいけません。すでに運用していた場合は速やかにアクセス制限を行いましょう

- アクセス制限の仕方は、本日の山口さんのチュートリアルを参考

- オープンリゾルバだと・・・

  - **DNS amp攻撃の踏み台に使われる（他人への迷惑）**

  - **DNS amp攻撃の踏み台にされることで負荷が増し動作が不安定になる**

  - **DNSキャッシュポイズニング攻撃がやりやすくなる**

アクセス制限する前提での構築・運用

- **組織がIPアドレスを獲得したら、DNSキャッシュサーバ担当へ連絡が行われてそのIPからのアクセス制限を解除する体制の構築が必要**
- **すでに、オープンリゾルバになってるキャッシュサーバにアクセス制限かけるのは非常に苦労します！（皆、大変苦労しています）**
  - 参考事例「オープンリゾルバ機能停止の取り組み」
    - NEC BIGLOBE小野さん / InternetWeek2013

https://www.nic.ad.jp/ja/materials/iw/2013/proceedings/d2/

ソースポートランダマイズ

- **キャッシュポイズニング攻撃からの最低限の防御**
- 必ずやりましょう
- 詳細は本日の山口さんのチュートリアルを参考

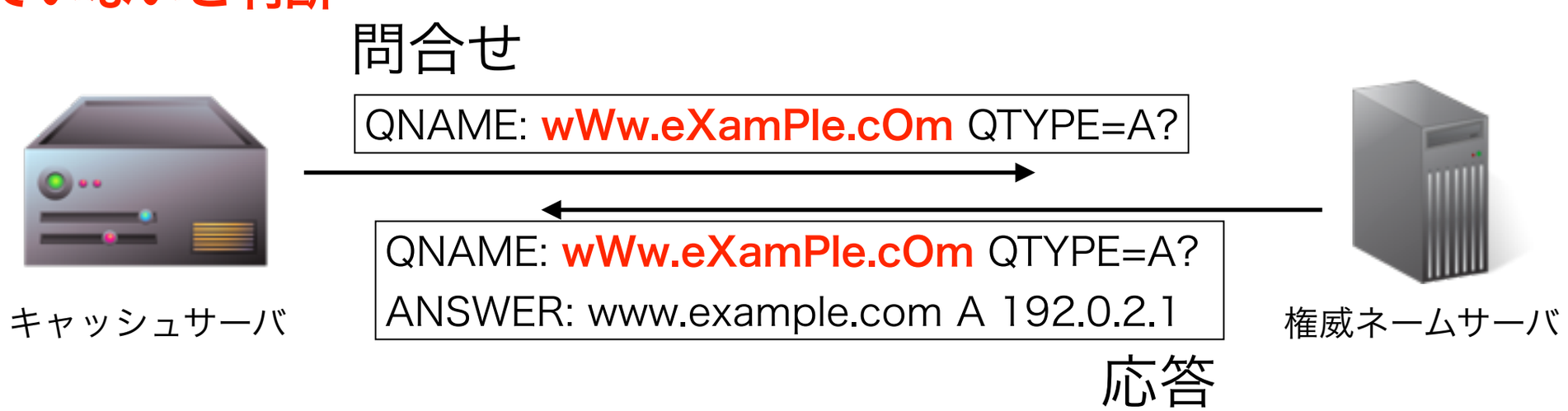
ポートランダマイズ以外の対策

- カミンスキー型攻撃の発表からすでに６年が経過
  - DNSサーバやネットワークの性能の向上等により、**従来のポートランダム化、TXIDランダム化だけでは今後不足する可能性もある**
- DNSSECで攻撃の検知と防御は可能だが、ここでは、敢えてDNSSEC以外の対策を紹介

dns-0x20

- ポートランダマイズ・TXIDランダマイズに加えて、**QUESTION SECTIONのQNAMEをランダム化**に利用しカミンスキー型攻撃等のブラインド攻撃をやりにくくするテクニック
  - DNSの問合せに対して、権威サーバは大文字小文字は無視して検索・応答するが、DNS応答のQNAMEは、DNS問合せのQNAMEをそのままコピーする仕様を利用

- **キャッシュサーバは**QNAMEをランダムに大文字小文字に変換**して権威サーバに送信。**権威サーバから全く同じQNAMEの応答が戻ってきたら応答が偽装されていないと判断



dns-0x20

- dns-0x20を利用できるキャッシュサーバは、フリーなものでは**Unboundのみ**

- use-caps-for-id: yes

  なんで0x20？→ ’A’ + 0x20 = ‘a’

dns-0x20

- 0x20の問題点
  - QNAMEを完全コピーせずに**全部小文字（または大文字）に変換して応答する権威サーバ**や、QNAMEが全部小文字でないと応答しない権威サーバもあり、そのような権威サーバが保持する**ゾーンが引けない等の問題が発生し得る**

ソースIPランダム化

- UnboundやPowerDNS recursor等の一部のキャッシュサーバでは、**クエリソースIPアドレスを複数指定可能**
- TXID・クエリソースポートに加え、**ソースIPもランダム化**されるためカミンスキー型**キャッシュポイズニング攻撃が成功する確率の低減**が期待できる
- IPv4アドレス枯渇のため、大量にIPアドレスを付与することはできないかもしれないが、**多数のユーザを収容するキャッシュサーバでは実施する価値あり**

Unboundでの設定例→
(ソースIP 8個でランダム化)

```
outgoing-interface: 192.0.2.1
outgoing-interface: 192.0.2.2
outgoing-interface: 192.0.2.3
outgoing-interface: 192.0.2.4
outgoing-interface: 192.0.2.5
outgoing-interface: 192.0.2.6
outgoing-interface: 192.0.2.7
outgoing-interface: 192.0.2.8
```

まとめ

- DNSが正しく動作しないと、インターネットのサービス利用に多大な影響が出る
- キャッシュサーバはDNSの重要な要素の一つだが、安定動作のためのノウハウがある
- パフォーマンスチューニング
- トラブルを防ぐための設定と環境
- トラブルへの備え
- セキュリティ

END

backup slides

IPフラグメントが届かない問題(補足)

- **DNSのメッセージサイズにありがちな誤解**
  - 「DNSは512バイトまではUDPを使う、それを超えるとTCPを使う。MTUは普通1500バイト近くあるので**UDPパケットのフラグメントは起こらないのでは？**」
  - 「DNSSECとかいうのを使うと512バイト以上でもUDP使うみたいだけど、うちのキャッシュサーバは**DNSSEC検証してないし関係ないよ**」
  - 「UDPで応答が得られなくても、**TCPでやり直すのでは？**」
- **キャッシュサーバ⇔権威サーバ間の通信に関しては、キャッシュサーバとしてBIND9やUnbound等の最近の実装を利用する限り上記はほぼ間違いです。フラグメントの問題は、ほぼ全てのキャッシュサーバの運用者に関係します。**

IPフラグメントが届かない問題(補足)

- **DNSメッセージサイズの実際**
  - キャッシュサーバ⇔権威サーバ間の通信は、ほとんどの実装でEDNS0が有効のため、**512バイト超えるDNSメッセージでもUDPを使用する。DNS応答が巨大になるとフラグメントされる**
  - 権威サーバ側でDNSSEC署名をされている場合、**キャッシュサーバでDNSSEC検証の有無に関わらず**DNSSEC署名つきの**巨大な応答が返る**
  - 権威サーバからUDP応答が得られない場合、**TCPで問合せをやり直すキャッシュサーバ実装もあるが、DNSの規格でそう決まっているわけではない。必ずそうするわけではない。**

Tips(1):ルートヒントの設定

- BIND9/Unboundはリリース時点の最新のルートヒント（ルート権威サーバのリスト）を内蔵しており、**ルートヒントを設定しなくてもDNSキャッシュサーバとしては正常に動作する**
  - ルートヒント（ルートネームサーバのリスト）は時々変更されるため、**明示的にルートヒントを与える設定にしておくのが良い**

- （別の意見）古いルートヒントを更新せずに使う人もいるし、どうせDNSキャッシュサーバのセキュリティ脆弱性が出て替えるのだから、**明示的にルートヒント与えないほうがいいのでは？**

Tips(2): UnboundでAAAA Filter

- **できないと（２年前に）言ったな、あれはウソだ**
  - private-address: ::/0
- 本来はDNS rebinding攻撃の防御として、ユーザへのDNS応答に含まれるプライベート・リンクローカルアドレスのA/AAAAレコードを落とすためのオプションだが、AAAAフィルタとして流用可能
- ただし、**BIND9のAAAA Filterとは少し動作が異なるので注意**
  - BIND9: AAAAのみ存在する名前(IPv6 onlyなサイト)はフィルタしない
  - 上記設定のUnbound: AAAAは常にフィルタしてしまう