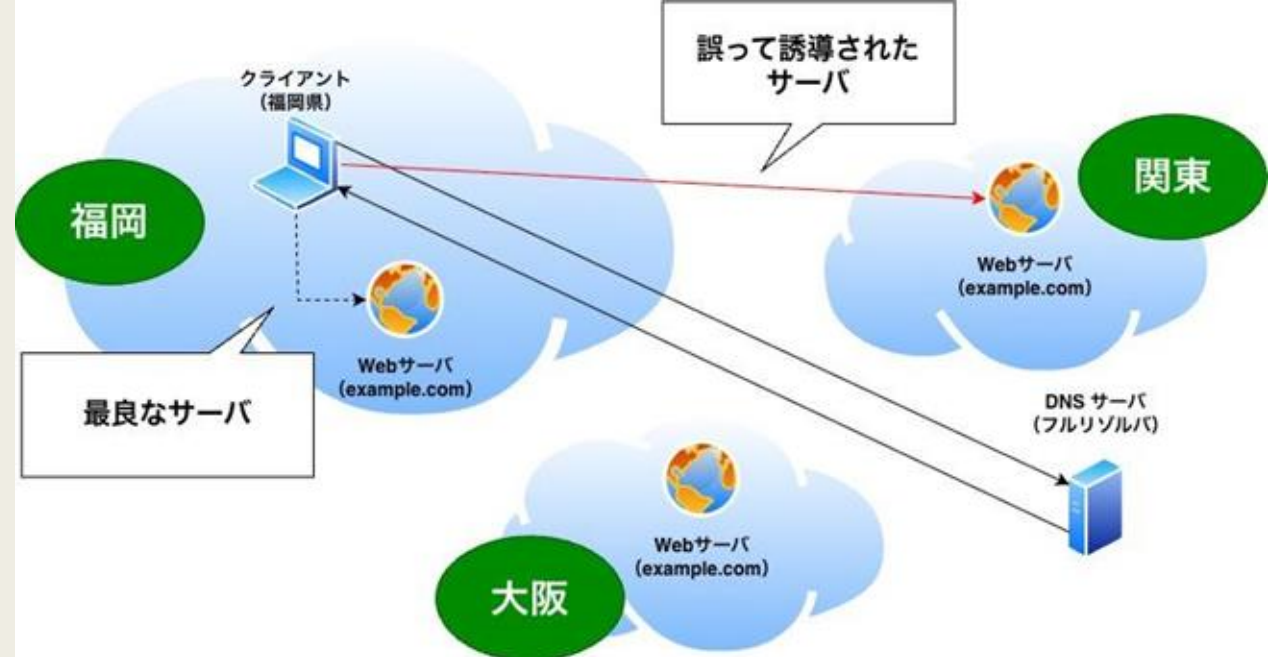


フルリゾルバにおける ECS 導入課題の 調査と解決策の提案

九州産業大学大学院 古賀 陽光

サーバ選択の問題



- 「フルリゾルバの位置 = クライアントの位置」とみなせない状況
 - フルリゾルバとクライアントが離れると、**最良なサーバ選択ができない**
- この問題が起きるケース
 - Public DNS（例：8.8.8.8）利用時
 - 基本的にフルリゾルバがクライアントと離れている[1]
 - ISP がフルリゾルバを集中管理している時[2]
 - 図：福岡のユーザーが関東のフルリゾルバを経由するケース

[1] F. Streibelt et al., "Exploring EDNS-Client-Subnet Adopters in your Free Time," IMC 2013

[2] 今泉 充司, "ECS(RFC 7871)を实际使おうと思うと困ること," JANOG56, 2025

EDNS Client Subnet (ECS)

- 「フルリゾルバの位置 = クライアントの位置」とみなせない時
 - クライアント側の情報を DNS 問い合わせに含める
 - 先述したサーバ選択の問題を解決
- ECS の登場
 - DNS 問い合わせ時にクライアント IP アドレスの一部を付加する技術[1]
 - ※本発表では IP アドレスの一部を「サブネット情報」と呼ぶ
- ECS を導入するメリット
 - フルリゾルバがクライアントと離れていても**最良なサーバ選択が可能**
- しかしフルリゾルバに ECS を導入するには課題が存在

ECS 導入における課題

- **フルリゾルバのメモリ使用量が増加する問題**
 - キャッシュエントリが増加
 - **メモリ使用量に与える影響の調査で見えた追加課題**
 - 平均名前解決時間やキャッシュヒット率への影響も確認
 - ECS 導入時には、メモリ使用量以外の課題も存在
- ECS 導入課題の整理と解決策の検討が必要

研究目的

- フルリゾルバのメモリ使用量が増加する問題の解決策を提案する
- 調査を通じて発見した ECS 導入時の課題を整理する
- 各課題について解決策を提案し、評価する

本研究で取り扱う課題

1. メモリ使用量の増加
2. ソフトウェアごとの挙動
3. ECS に関する情報の観測性
4. 権威サーバの想定と異なるキャッシュ
5. プライバシーの懸念

本研究で取り扱う課題

1. メモリ使用量の増加
2. ソフトウェアごとの挙動
3. ECS に関する情報の観測性
4. 権威サーバの想定と異なるキャッシュ
5. プライバシーの懸念

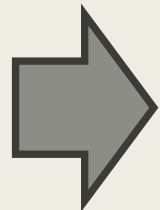
課題① メモリ使用量の増加 ECS によるキャッシュの細分化

- フルリゾルバでは、サブネット情報も含めてキャッシュする
 - 同じドメイン名でも、サブネット情報によって応答が変わるため
 - サブネット情報（プレフィックス）ごとにキャッシュが分かれる
 - キャッシュエントリが増加し、メモリ使用量が増加
- どの程度増加するかを測定

ECS 無し

問い合わせ

example.com



キャッシュエントリ

IP アドレスのみ

ECS を含む

問い合わせ

example.com + subnet=49.128.1.0/24
example.com + subnet=60.67.1.0/24
example.com + subnet=110.41.1.0/24



キャッシュエントリ

IP アドレス + 49.128.1.0/24
IP アドレス + 60.67.1.0/24
IP アドレス + 110.41.1.0/24

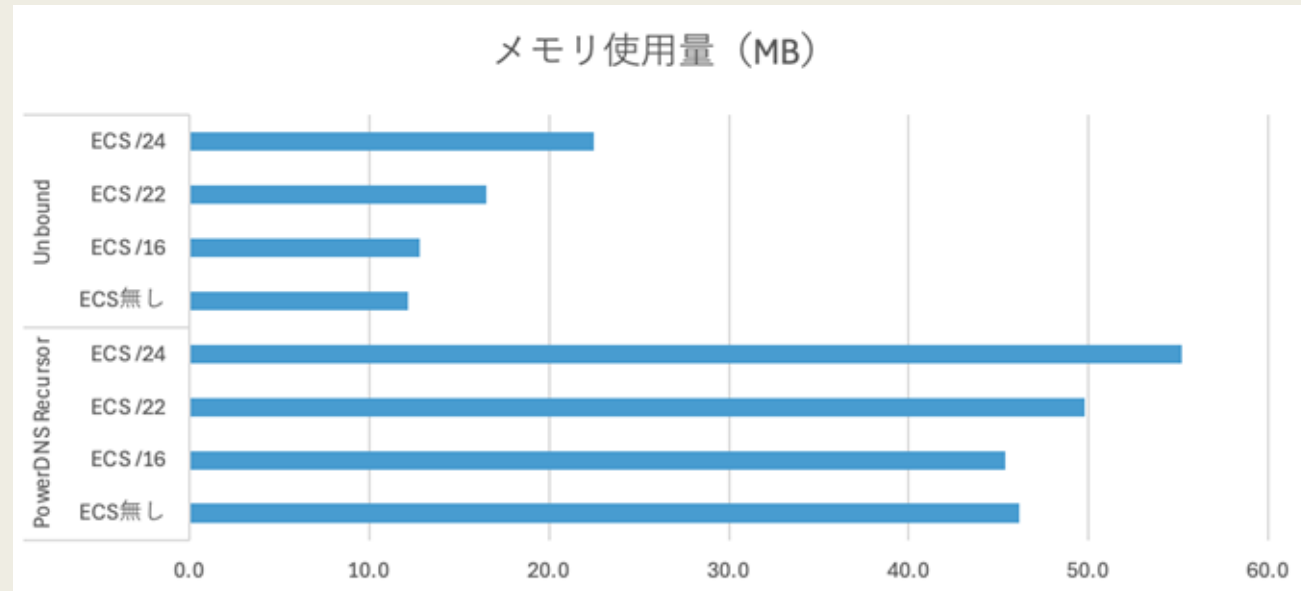
別のキャッシュとして扱われる

課題① メモリ使用量の増加 これまでの調査（1）

- 2種類の OSS フルリゾルバでメモリ使用量を測定
 - テストデータで名前解決を実施後、プロセスの実メモリ使用量を測定
 - Unbound 1.21.0, PowerDNS Recursor 5.2.2
- サブネット情報の長さを変更して影響を比較
 - ECS 無し、ECS /16, /22, /24 で比較
 - 各フルリゾルバにテストデータで問い合わせし、メモリを測定
 - 5回実験した平均値

課題① メモリ使用量の増加 これまでの調査（２）

- サブネット情報が長いほど、メモリ使用量が増加
 - ECS /16 より ECS /22, /24 の方が増加
 - ECS /16 と ECS /24 で、**約 1.2～1.8 倍**の差が見られた



課題① メモリ使用量の増加 解決策の提案

- **方針：ECS の利用を効果が見込める問い合わせに限定**
 - 関連研究：返す IP アドレス数が 1 つしかなく、ECS の効果が無いと思われるドメイン名が 62% を占める[1]
 - その結果、キャッシュが不要に細分化されメモリ使用量が増加
- **実現案：問い合わせをECS 対応・非対応系に振り分ける**
 - ECS の効果がある問い合わせ：ECS 対応フルリゾルバへ転送
 - それ以外：通常のフルリゾルバで処理

→ ECS キャッシュを削減し、メモリ使用量の増加を抑える

本研究で取り扱う課題

1. メモリ使用量の増加
2. **ソフトウェアごとの挙動**
3. ECS に関する情報の観測性
4. 権威サーバの想定と異なるキャッシュ
5. プライバシーの懸念

課題② ソフトウェアごとの挙動

- 同じ ECS 設定でも、フルリゾルバごとに挙動が異なる
 - Unbound と PowerDNS Recursor で異なる挙動を確認
 - キャッシュ関連の処理や ECS 関連の挙動に差が存在
- ECS 導入時には、利用するソフトウェアの挙動の確認が必要

観点	Unbound	PowerDNS Recursor
メモリ使用量	サブネット情報が長くなるほど増加	ECS /16 より ECS 無しが多い
ECS応答	ECS 付き問い合わせに ECS を返す	一部条件で ECS を返さない (RFC に違反)
キャッシュ確認	cache_lookup で確認可能 (ドメイン名ごと) ※1.24.0以降	dump-cache で確認可能 (キャッシュ全体)
観測方法(dnstap)	クライアント側・権威側を記録可能	権威側のみ記録可能

本研究で取り扱う課題

1. メモリ使用量の増加
2. ソフトウェアごとの挙動
3. ECS に関する情報の観測性
4. 権威サーバの想定と異なるキャッシュ
5. プライバシーの懸念

課題③ ECS に関する情報の観測性

■ ECS 導入時に確認したいこと

- サブネット情報によって応答が変わるか
- フルリゾルバがどのように問い合わせ・応答を処理したか
- フルリゾルバが実際にキャッシュしたサブネット情報

■ dig / nslookup コマンドで見える情報は限定的

- 最終的な応答に含まれる ECS 情報のみ確認可能
- フルリゾルバがどんな応答をもらった・どうキャッシュしたかは不明

→ ECS 導入時の検証・評価が難しくなる

本研究で取り扱う課題

1. メモリ使用量の増加
2. ソフトウェアごとの挙動
3. ECS に関する情報の観測性
4. **権威サーバの想定と異なるキャッシュ**
5. プライバシーの懸念

課題④ 権威サーバの想定と異なるキャッシュ

- 権威 DNS サーバは、サブネット情報をもとに応答する
- フルリゾルバが粗い粒度（例：/16）で問い合わせると、細かい粒度の応答（例：/24）が活用されない場合がある[1]
- その結果、権威 DNS サーバが用意した応答の一部が使われず、最良なサーバ選択に影響する可能性がある



[1] 今泉 充司, “ECS (RFC 7871) を実際使おうと思うと困ること,” JANOG56, 2025.

本研究で取り扱う課題

1. メモリ使用量の増加
2. ソフトウェアごとの挙動
3. ECS に関する情報の観測性
4. 権威サーバの想定と異なるキャッシュ
5. プライバシーの懸念

課題⑤ プライバシーの懸念

- サブネット情報が長いほど、利用者側の情報が細かく伝わる
 - 一方で、短くしすぎると最良なサーバ選択の効果が弱くなる可能性がある
 - 適切なサブネット情報の長さは、フルリゾルバの運用構成によって異なる
- ECS を使う対象や、送るサブネット情報の長さを制御することは重要

今後の課題

■ 課題①（メモリ使用量）

- 提案した振り分け手法を実装し、効果を評価

■ 課題②～⑤

- 引き続き調査を進める
 - ソフトウェアごとの挙動の要因調査（課題②）
 - ECS の観測性についての調査（課題③）
 - 権威サーバの想定と異なるキャッシュによる影響の調査（課題④）
 - サブネット情報の長さによってプライバシーに与える懸念の調査（課題⑤）

まとめ

- ECS は、フルリゾルバがクライアントと離れている場合でも、最適なサーバ選択を可能にする技術である
- 本研究では、フルリゾルバへの ECS 導入時に生じる課題を整理した
- 調査の結果、メモリ使用量の増加だけでなく、ソフトウェア差・観測性・キャッシュ・プライバシーの課題を確認した
- 今後、メモリ使用量が増加する問題については ECS が必要な問い合わせのみを対象とする手法を検討し、提案手法を評価する
- その他の課題については現状を調査し、解決策を提案していく