

予測型DNSプリフェッチに関する考察

2026年6月26日
NTTドコモビジネス株式会社

末松 慶文

- ・ 自己紹介と背景
- ・ 名前解決待ちによる遅延を低減する技術
- ・ フルリゾルバにおけるPrefetchとは
- ・ 従来型Prefetchの動作 (BIND9)
- ・ 予測型Prefetchの動作 (Knot Resolver)
- ・ 予測型Prefetchの予測モデル
- ・ 予測型Prefetchの評価

■ 免責事項、その他

- ・ 発表内容は個人の見解であり、所属する組織の見解を示すものではありません。

- ・ 名前：末松 慶文
- ・ 所属：NTTドコモビジネス株式会社 / DNSOPS.jp幹事
- ・ 仕事：ISPとして提供しているフルサービスリゾルバ(キャッシュDNS)と
権威DNSの設計および開発（回線サービス名：OCN）

- ・ 今回の発表について

フルリゾルバにおける名前解決待ちによる遅延を低減する技術として複数のPrefetch方式がある

従来型Prefetch：TTLが残り少ないキャッシュ済レコードをバックグラウンドで再取得

予測型Prefetch：再度必要と見込まれるレコードを予測し、事前に取得

Prefetchの有効性

大規模環境では、従来型Prefetchの発動条件を満たしやすいために有効に機能しやすい

一方、小規模環境では、発動条件を満たしにくいため、従来型Prefetchは有効に機能しづらい

小規模な環境でも有効に機能すると考えられる予測型のPrefetchに着目し、その有効性を評価する

■ 免責事項、その他

- ・ 発表内容は個人の見解であり、所属する組織の見解を示すものではありません。

名前解決待ちによる遅延を低減する技術

つながる。驚きを。幸せを。

 **docomo Business**

アプリケーション

例：Webブラウザによる、投機的読み込み(Speculative loading)による応答遅延低減

- 名前解決の事前取得
- 接続ハンドシェイク(DNS + TCP + TLSなど)を事前に実行

DNS

例：スタブリゾルバによる、期限切れ応答の暫定的な利用による応答遅延の低減

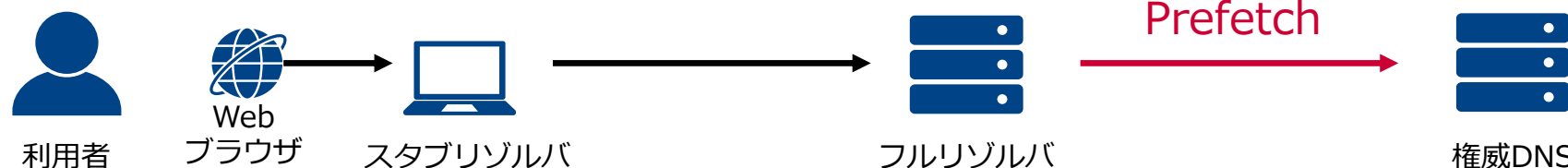
- 暫定的な利用に並行して名前解決を実行し、必要があれば補正する

Optimistic DNS: <https://www.ietf.org/archive/id/draft-gakiwate-dnsop-optimistic-dns-00.html>

例：フルサービルリゾルバによる、名前解決の事前取得による応答遅延の低減

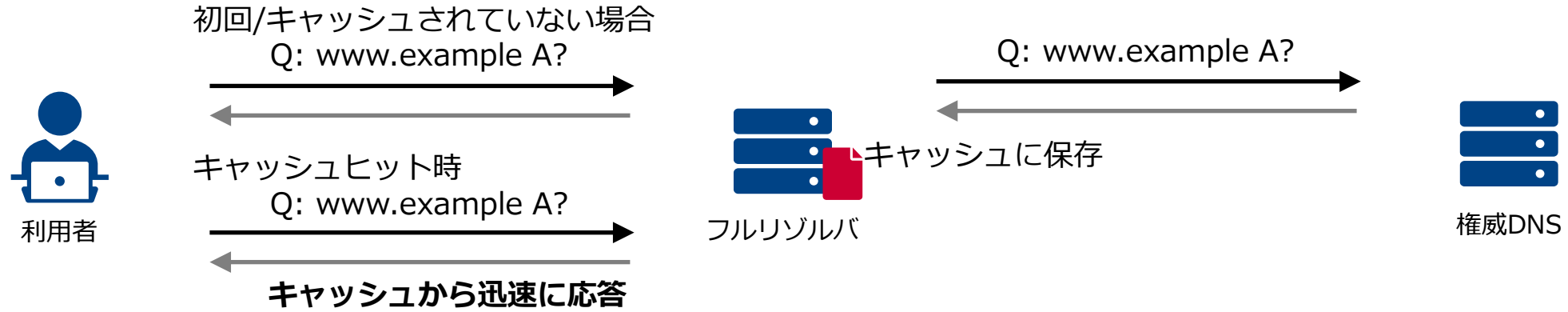
- 従来型のPrefetch (BIND9, Unbound, Knot Resolverなど)
TTLが残り少ないレコードをバックグラウンドで事前に取得

- 予測型のPrefetch (Knot Resolver)
再度必要と見込まれるレコードを予測し、事前に取得

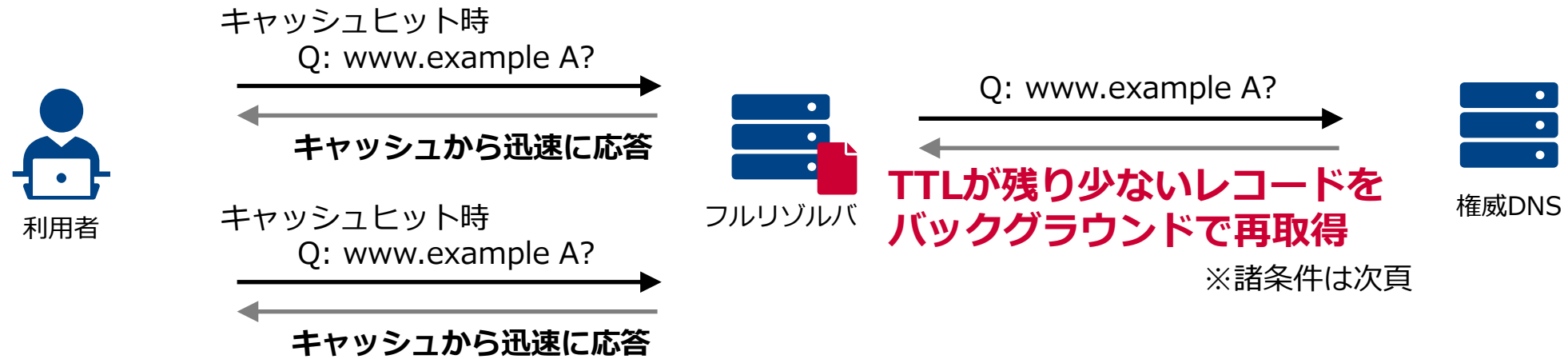


フルリゾルバにおけるPrefetchの動作

通常の名前解決動作



Prefetch有効時の名前解決動作(従来型)



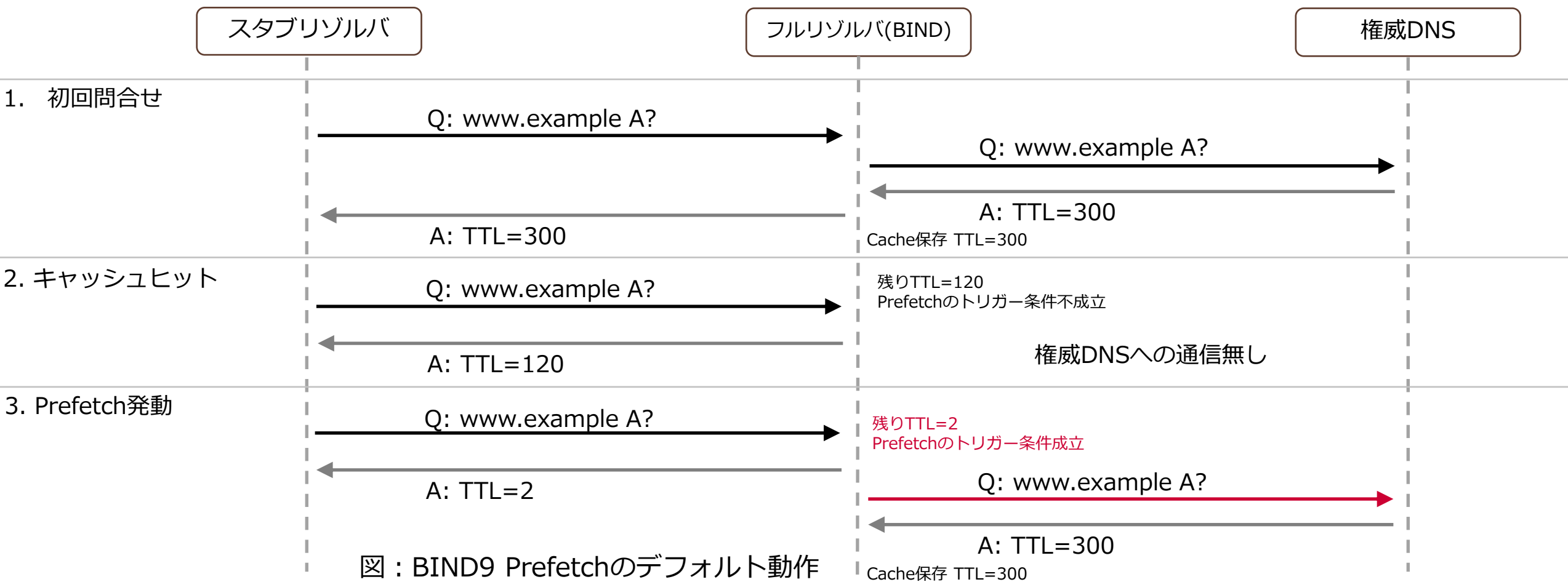
事前を取得し、キャッシュをホットに保つことで、応答遅延を軽減

従来型Prefetchの動作 (BIND9)

つながろう。驚きを。幸せを。

 **docomo Business**

- TTLが残り少ないキャッシュ済レコードに問い合わせが来た際、バックグラウンドで再取得
- デフォルト：残りTTL2秒以下でprefetch発動、オリジナルTTL9秒以上のRRsetをprefetch対象とする



小規模環境では、発動条件を満たしにくいいため、従来型Prefetchは有効に機能しづらいと考えられる

予測型Prefetchの動作 (Knot Resolver)

つながる。驚きを。幸せを。

 **docomo Business**

- 過去の利用パターンをもとに再度必要と見込まれるレコードを予測し、事前を取得
- 予測モデルは、問い合わせのサンプリング結果をもとに周期性な再出現を推定する仕組み

プロトタイプのため、商用環境や高トラフィック環境では非推奨

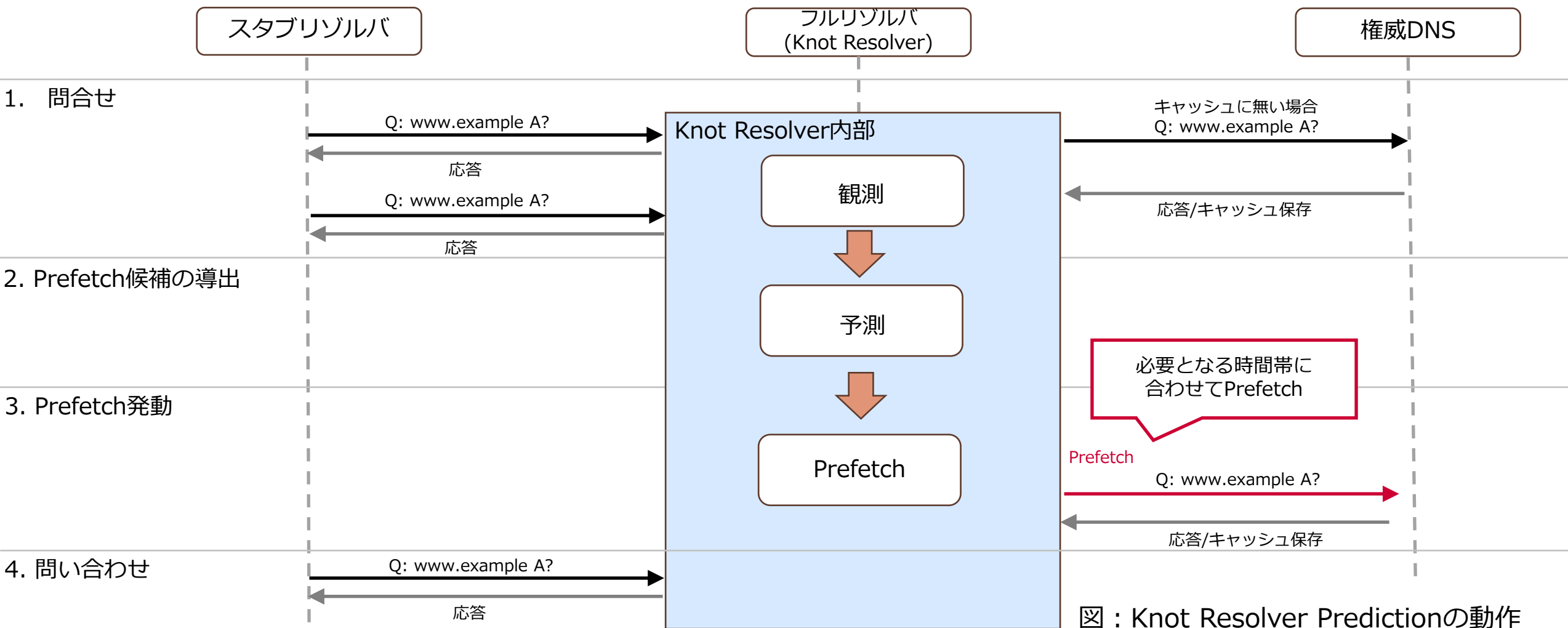


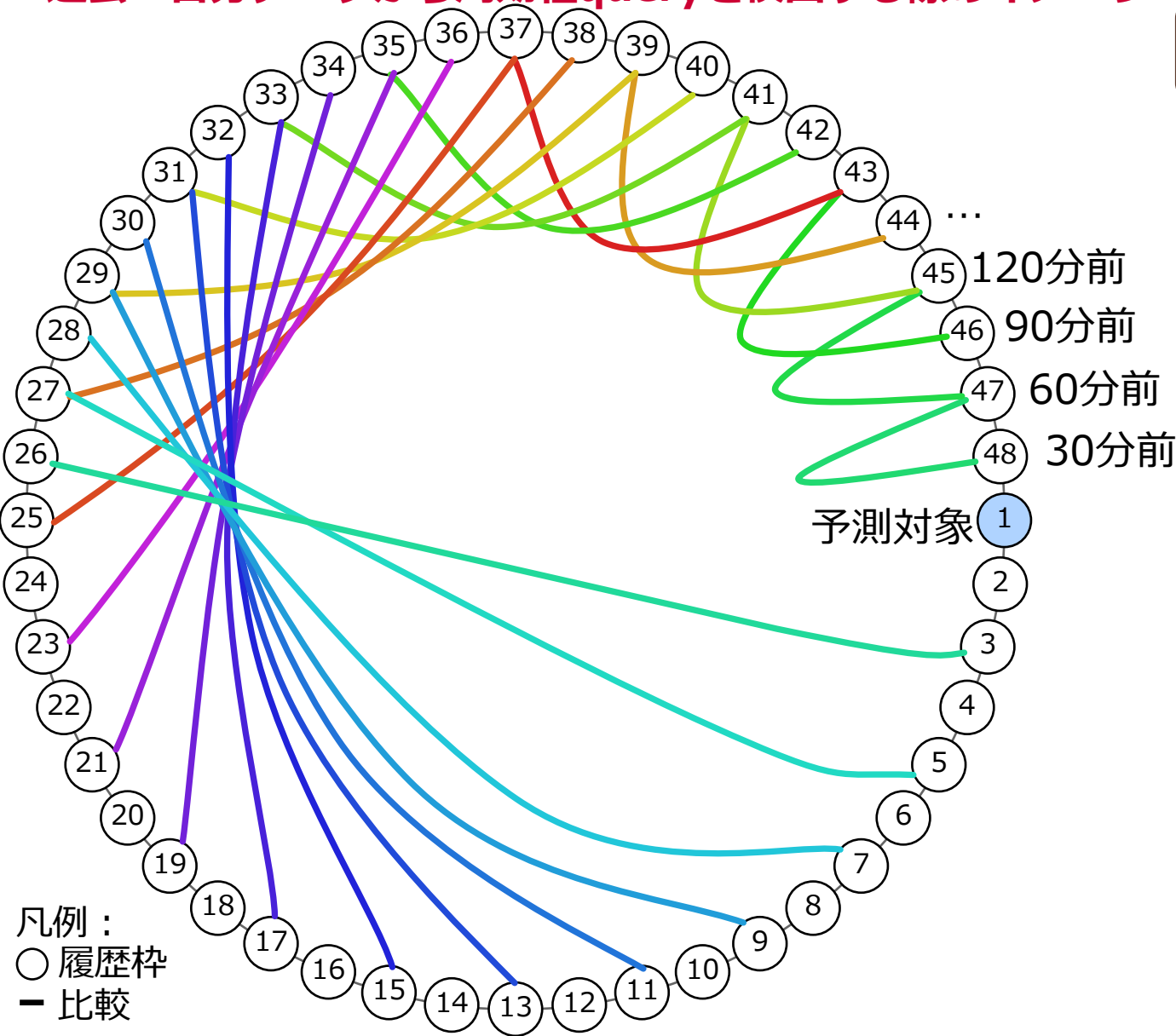
図 : Knot Resolver Predictionの動作

予測型Prefetchの予測モデル(Knot Resolver)

過去一日分データから周期性queryを検出する際のイメージ



- 観測
- ・ queryをサンプリングし、履歴枠に記録
- 予測
- ・ 過去の履歴枠を比較し、周期的に再出現が見込まれるqueryをPrefetch対象として予測



周期性	比較	見込み
30分	48と47	1で再出現
1時間	47と45	1で再出現
1.5時間	46と43	1で再出現
...		
11時間30分	26と3	1で再出現

前提条件：
30分の履歴枠×48枠 = 24時間を周期
(window = 30m, period = 48)

予測型Prefetchの有効性は予測精度に依存するため、実環境での評価が必要

予測型Prefetch 評価内容

つながろう。驚きを。幸せを。



評価目的

予測型Prefetchにより、名前解決時の応答性能が改善するかを「レイテンシ」「キャッシュヒット率」「予測機構の動作」の観点から評価する

評価仮説

予測型Prefetchの予測機構が動作し、Prefetchが行われることでキャッシュヒット率が向上し、レイテンシが向上

評価対象

予測型Prefetch有効(15分, 24枠=6時間)、予測型Prefetch有効(30分, 48枠=1日)、予測型Prefetch無効

評価条件・データセット

環境 : Ubuntu 26.04 LTS(VM), Knot Resolver6.3.0

データ : 自宅フルリゾルバの実トラフィック

収集 : Prometheus, Grafana 5分粒度

評価項目

予測機構の動作 : 予測サイクル(epoch), 学習量(learned), 先読み候補数(queue)

応答性能の向上 : キャッシュヒット率, p95・p99 レイテンシ

※クライアント要求とResolver内部要求を含む

品質影響 : SERVFAIL, CPU負荷

予測型Prefetch 評価結果

つながる。驚きを。幸せを。



評価結果

予測機構とPrefetchの動作により、キャッシュヒット率が向上とレイテンシが低下する想定であったが、双方とも改善は確認できなかった。また、予測設定へ変更(30分,48枠=1日)した結果、予測量の増加は確認できたが、応答性能の改善は確認できなかった。

指標		有効 15分×24枠	有効 30分×48枠	無効
予測機構	学習量 epoch内peak / 中央値	17 / 1.13	40 / 1.33	N/A
	先読み候補数 epoch内peak / 中央値	4 / 0.27	68 / 2.77	N/A
応答性能	Cache hit率 期間全体 / 5分中央値 ※	82% / 26%	75% / 27%	83% / 29%
	P95 latency 期間全体 / 5分中央値 ※	79ms / 214ms	115ms / 206ms	77ms / 203ms
品質影響	SERVFAIL率	0.33%	0.48%	0.42%
	CPU負荷 (OS)	2.74%	2.73%	2.73%

※短時間かつ大量・高いhit率のトラフィック影響で期間全体の値が良好に見える

考察

予測型Prefetch 30分×48枠では、学習量と先読み候補数が増加したことから予測機構は積極的に動作したことが確認できた。一方、キャッシュヒット率とレイテンシの向上は確認できなかった。この要因は、結果と実際の問い合わせクエリーが一致しなかった、あるいは、予測に基づくPrefetch前にすでにキャッシュされていたことが要因と推測される。今後、より詳細な分析および分析結果に基づく予測機構の設定チューニングを行いたい。

つながう。驚きを。幸せを。

