

パケット監視に基づく DNS キャッシュサーバ拡張システムの試作

義田博一[†] 山内雪路[‡]

^{†‡} 大阪工業大学大学院情報科学研究科

1 はじめに

昨今、インターネット上での Web サービスにおいて、1 ページの描画に必要な時間が増加している。これはページ中に含まれるコンテンツの種類が増加し、それらのコンテンツが複数のドメインを参照することが原因である。そこで DNS による名前解決の高速化が注目されている。DNS には一度名前解決したドメイン名に対する資源レコードをキャッシュできる仕組みがあるが、資源レコードの TTL の制限により、必ずしも常にキャッシュを活用できるとは限らない。

そこで本研究では、DNS キャッシュサーバのキャッシュ効率 (Hit 率) を上げるために、既存の DNS キャッシュサーバを拡張させるシステムを提案する。DNS キャッシュサーバを拡張する研究として、適切なミラーサーバを選択できるように拡張するもの [1] や、最近になってサービスがリリースされた Google Public DNS [2] などが挙げられる。これらの研究における DNS キャッシュサーバの拡張方法は、キャッシュサーバのコードを新規に作成し、これに拡張機能を盛り込むという共通の特徴を持っている。本研究では、既存の DNS ソフトウェアに一切の手を加えずにキャッシュ永続化を図る拡張機能を実現する手法を提案する。

2 拡張システムの提案手法と実装方法

本拡張システムは全て Ruby および Ruby の拡張ライブラリで作成している。図 1 に本拡張システムの全体構成を示す。

2.1 パケット監視による拡張システム

DNS キャッシュサーバ上で DNS パケットを監視し、拡張機能部で独自にキャッシュテーブルを構築し、資源

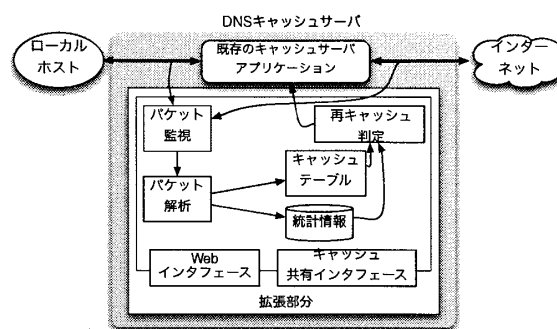


図 1 システム全体の構成図

レコードを管理する。TTL が超過した資源レコードを、再キャッシュするか否かを後述のアルゴリズムで判定し、既存のキャッシュサーバアプリケーションに問い合わせることでキャッシュを更新し永続化できる。既存の DNS キャッシュサーバに全く手を加えていないので、バグを新たに埋めこんだり、セキュリティホールを作ってしまうこともない。また、本システムが不具合によりダウンしたとしても、キャッシュサーバには影響を一切与えることなく、名前解決サービスを中断することがない利点があると考えられる。

2.2 問い合わせの頻度による再キャッシュの判定

TTL が超過したレコード全てを対象に再キャッシュすると、計算機資源を喰いつぶしてしまうほか、無駄な DNS 問い合わせを多数発生させることになり適切ではない。そこで、本拡張システムの再キャッシュ方法に、問い合わせの頻度から TTL を超過した資源レコードを再キャッシュするかどうか判定する手法を提案する。

再キャッシュの可否判断は、2 段階のステップで判定する。この判定にはその資源レコードの参照情報と問い合わせの頻度を特徴として使用する。

- TTL が超過するまでにそのレコードが再び参照されているか
 - － 参照されていればそのレコードを再キャッシュする
 - － 参照されていなければ問い合わせの頻度から再

Extendend System of DNS Cache Server based on Packet Monitoring

Hirokazu YOSHIDA[†], Yukiji YAMAUCHI[‡]

^{†‡}Osaka Institute of Technology

573-0196, Osaka, Japan

[†]h.yoshida@cosymlab.is.oit.ac.jp [‡]yamauchi@is.oit.ac.jp

度問い合わせされる確率を求めて再キャッシュするか判定する

再度問い合わせが発生する確率は、ある時間帯に出現する確率 (P_{zone}) およびある日に出現する確率 (P_{day}) の 2 つの出現確率から求める。

$$P_{zone} = \text{問い合わせが 1 日に出現する時間帯数} / 24$$

$$P_{day} = \text{問い合わせが出現した日数} / \text{経過日数}$$

この 2 つの確率を掛けあわせ、ある日の時間帯に出現する確率 (P_{total}) を算出し、この確率で再キャッシュを試行する。

$$P_{total} = P_{zone} P_{day}$$

2.3 複数台環境下におけるキャッシュの共有

本拡張システムは、DNS キャッシュサーバを複数台管理している場合においても、システム間で通信しキャッシュを共有できる。DNS コンテンツサーバにおけるマスターとスレーブの概念を取り込み、マスター-スレーブ間でキャッシュを共有できる。

3 拡張システムの性能評価

本拡張システムを、DNS キャッシュサーバ 1 台で稼働させた場合の性能を評価する。大阪工大情報科学部 (在学生 2000 名) の Web プロキシサーバのアクセスログを基に性能を評価した。そのアクセスログを 5 日分用いて、大学内に流れる疑似的な DNS 問い合わせとして拡張システムを稼働させ、再キャッシュを有効にした場合と無効にした場合についてログの時刻に基づいて DNS キャッシュサーバに問い合わせた。評価する項目として、Hit 率および再キャッシュした回数を計測し、その結果を図 2、図 3 に示す。

図 2 より、資源レコードを再キャッシュしない (no) 場合より、再キャッシュする (all, refprob) 方が、Hit 率がより高くなっていることがわかる。このことから、資源レコードを再キャッシュすることがキャッシュ効率を高めるために有効であるといえる。

全て再キャッシュ (all) した場合と頻度に基づいて再キャッシュ (refprob) した場合について比較すると、all の方が若干高い Hit 率を維持できるが、refprob の場合は問い合わせの頻度から、必要と判定した資源レコードを徐々に再キャッシュできるようになっていると言える。加えて、図 3 より refprob は all に比べると再

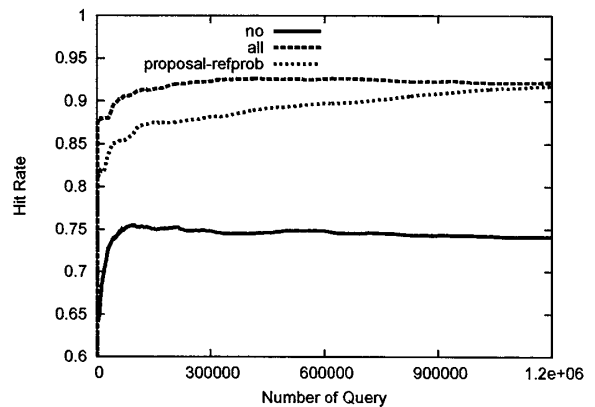


図 2 Hit 率の平均値

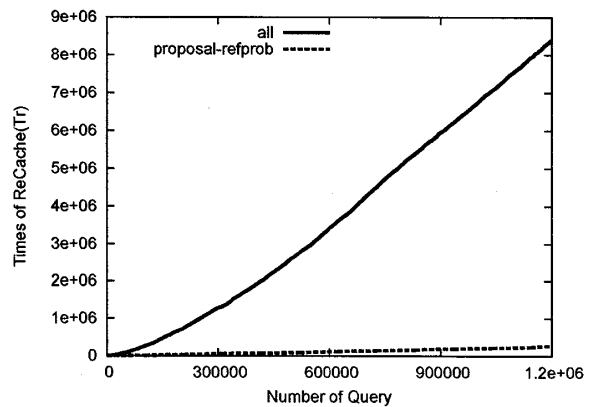


図 3 再キャッシュ回数の平均値

キャッシュ回数 (Tr) が非常に少ない。また、異なるログについても同様に性能を評価した結果、同様な結果が得られた。このことから、refprob はサーバやインターネットに負荷をかけることなくキャッシュの効率を著しく向上できるものと考えられる。

4 まとめ

本研究では、DNS パケットの監視に基づいて DNS キャッシュサーバを拡張させるシステムを提案、実装した。評価実験により問い合わせの頻度から再キャッシュすることで、DNS キャッシュサーバのキャッシュ効率を負荷を抑えつつ上げられることを示した。

参考文献

- [1] 横田 裕思, 木村 成伴, 海老原 義彦, “DNS フィルタ方式によるミラーサーバ選択法の提案と実装”, 情報処理学会論文誌, Vol.44, No.3, 2003
- [2] Google Public DNS, <http://code.google.com/speed/public-dns>