

DNS キャッシュ寿命の戦略的決定

清水 奨

NTT ソフトウェア研究所

1995 年 10 月 26 日

インターネットの分散データベースである DNS(ドメイン名システム)では、問い合わせ負荷の軽減を目的としてキャッシュが用いられる。キャッシュの寿命(TTL)はサーバが秒単位で指定するが、従来この値は経験的に決められる事が多かった。本稿では、TTL を変化させた場合の問い合わせ数の変化を分析し、問い合わせの傾向にあわせた TTL を動的に指定する事で、従来ほとんど使われる事の無かった極めて短い TTL 値がデフォルトとして使用可能である事を示す。NTT で実際に使用中の對外 DNS サーバの TTL 値を 6 日、12 時間、1 時間、5 分と変化させ、他の DNS サーバからの問い合わせ内容をロギングした。問い合わせの集中するデータの上に長い TTL 値を与えることで、デフォルト TTL を 5 分としてもサーバ負荷を抑える事が出来る。最適 TTL 値の決定手法を紹介する。

1 はじめに

インターネットの一般化に伴い、DNS(Domain Name System)[1]の重要性も増してきている。DNS はインターネットワイドな一種の分散データベースであり、主にホスト名から IP アドレスを引くのに用いられる。問い合わせの集中を防ぐため、ネームサーバにはキャッシュが備わっている。

キャッシュの寿命は経験的に「数日」が推奨され [1][2]、利用されている。これは、DNS が「ドメイン名やマシン名はそれ程頻繁に変更されるものではない」という考えに基づいているためである。実際にキャッシュの寿命は、通常「数日間」が望ましい。

しかしネットワークの移設や廃止など、DNS レコードが頻繁に変更されるような場面では、長いキャッシュ寿命はトラブルの元である。言うまでもなく、変更前の古いデータがあちこちにキャッシュされ、新データがなかなか反映されない。そこで TTL を短く設定するのだが、どの程度短く出来るのか、短くすると何が起るのか書かれた文献は少ない。

本論文では、ネットワークの変更直後など、DNS レコードの変更が比較的多く発生するような状況において、TTL の値をできるだけ短くする方法について

のべる。NTT のネームサーバを例にした実際の統計データから、実験的な検討を行った。

以下二章では DNS のキャッシュについて述べ、三章ではキャッシュ寿命(TTL...Time To Live)を単純に変化させたときの問い合わせの傾向変化について述べる。さらに四章で問い合わせの性質に注目した TTL の決定法を紹介し、五章で今後の課題を述べ、結ぶとする。

2 DNS におけるキャッシュ

DNS についての詳細は書籍 [3] に詳しいが、ここではそのキャッシュについて簡単に触れておく。

図 1 に、ホスト名からアドレスを解決するプロセスを例に、DNS の問い合わせの発生から回答の受け取りまでの流れを示す。

ユーザがホスト名をドメイン表記で行うと、マシンのリゾルバルーチンが IP アドレスを求めるパケットを生成し、ローカルなネームサーバに送る(図中 1)。ローカルネームサーバは、ルートから始めて順に階層構造の下の方へと問い合わせを行き(2,3)、最後に探している相手先のネームサーバにたどり着き(4)、回答を得る(5)。そして、その回答をリゾル

東京都武蔵野市練町 3-9-11, M9-311A NTTソフトウェア研究所 広域コンピューティング研究部, mail: shimiz@ntt-20.ntt.jp

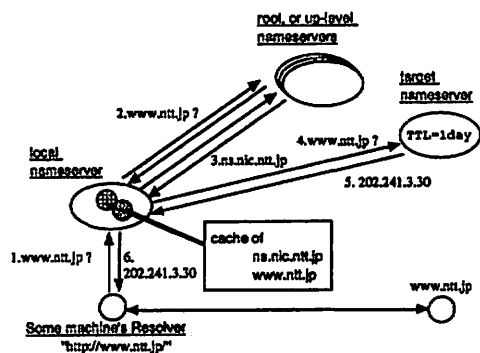


図 1: DNS の問い合わせデータの流れ

バにわたす (6)。

この一連の問い合わせの流れで、ローカルネームサーバには相手先のネームサーバ自身の IP アドレスと、今回問い合わせた内容の二つのレコードがキャッシュされる¹。このキャッシュの寿命は、相手先のネームサーバの TTL(Time to Live) レコードで指定される(図中,1day)。

次回同じ内容の問い合わせがあれば、相手のネームサーバまで聞きに行かずキャッシュを見て答えを返すことができる。また、同じ内容でなくとも、同じドメインの問い合わせであれば、ルートネームサーバに聞きに行かずに直接相手先ネームサーバに問い合わせることができる。このようにして DNS のキャッシュは各ネームサーバの負荷を軽減している。

自分の管理しているネームサーバのキャッシュ寿命(TTL)を長くするという事は、そのネームサーバに問い合わせしてきた外部のネームサーバにそれだけの長い時間データを保持してもらうということである。外部のネームサーバがキャッシュしたデータは、寿命が尽きるか、サーバをリスタートしないと消えることはないの、ネットワークの移設などでレコードの内容が頻繁に変更される場合はできるだけ TTL を小さくしたいということになる。

TTL の長短と、一般に考えられているそれぞれの長所短所を表にまとめると、表 1 のようになる。次章では、この現状認識のうち特に短い TTL の短所としてあげられているサーバ及びネットワークの負荷について、実験を通じて再確認する。

¹上位ドメインのネームサーバの IP アドレスもキャッシュされるが、ここでは特に触れない。

表 1: TTL の長短と特性

項目	長い TTL	短い TTL
変更の伝播速度	遅い (×)	早い (○)
外部サーバのキャッシュ	多い (×)	少ない (○)
関係サーバの負荷	軽い (○)	重い (×)
ネットワーク負荷	軽い (○)	重い (×)

3 TTL の変化と問い合わせの変化

この章では、TTL を小さくした場合、大きくした場合の問い合わせの様子の変化を実際のデータを用いて図示する。いずれのグラフも、同一の設定で一週間運用し、一日当たりの平均値を求めてプロットしている。また、同じ内容の問い合わせが同じネームサーバから来る時間間隔のパターンから、TTL の有効性について検討する。各々のデータは NTT で稼働中のネームサーバの一つ(データレコード 1100 余、サブドメイン指定 23 箇所)を用いて行った。

3.1 TTL の変化と総問い合わせ数の変化

図 2 に、TTL を変化させた場合の全問い合わせ数(ただし、リゾルバからの問い合わせは除く、以下同じ)²の変化を図示する。全問い合わせ数は、表 1 のネットワーク負荷に対応するものである。また、ネームサーバに用いられているマシンの性能によっては、サーバの負荷にも関係する。

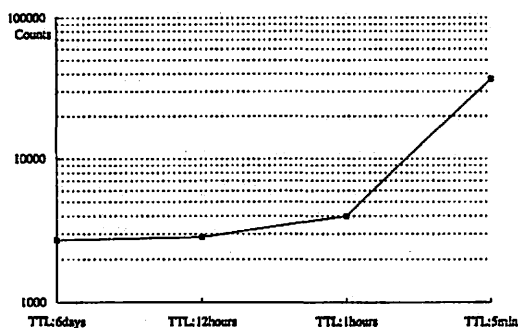


図 2: TTL の変化に伴う全問い合わせ数の遷移

図では横軸 TTL の長さを 6 日、12 時間、1 時間、5 分のように 12 倍のスケールで長い順に並べてあり、

²リゾルバの問い合わせはキャッシュの影響を受けないので、カウントから除いている。

縦軸にはネームサーバからの問い合わせ数を log スケールで示している。TTL が 5 分のときには極めて多くの問い合わせがきていることがわかるが、1 時間より長い場合、特に 12 時間と 6 日間の場合の問い合わせ数がほとんどかわらないことが確認できる。

この結果から、TTL はある程度長くしておけば、それ以上長くしても問い合わせ数において大きな差はないと推測される。

3.2 TTL の変化に伴う重複問い合わせの比率の遷移

図 3 は、同じ内容の問い合わせが全体に占める比率を、TTL を前例と同様に変化させてプロットしたものである。この図は、表 1 の評価項目とは直接関係しないが、外部のキャッシュがどの程度有効に利用されているかを知る目安になる。

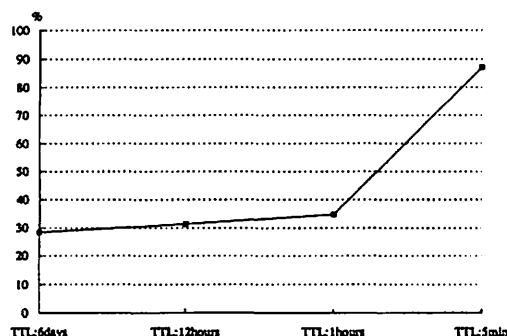


図 3: 重複問い合わせの比率

この図においても特異なのは TTL が 5 分と極めて短い場合であり、他の TTL 値の場合は大きな差は認められない。TTL が 5 分の場合の著しい重複は、単純に TTL を短くすることが効率の面でかなり悪影響を及ぼすことを示している。

3.3 TTL 固定での問い合わせ間隔

図 4 は、同じネームサーバから同じ内容の問い合わせが来る時間間隔をカウントしたものである。計測時間は 24 時間である。この図は、TTL 指定とは関係のない問い合わせがどれくらいあるかを示し、なぜ TTL が 6 日の場合と 12 時間の場合で大きな差がないかを説明するものである。

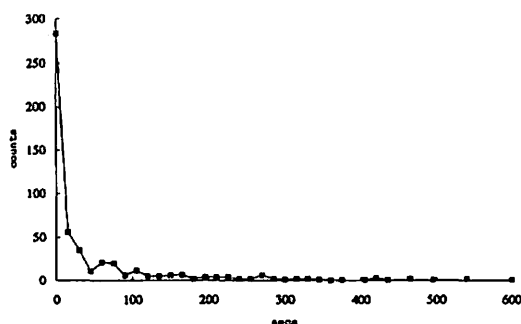


図 4: TTL=12hours, 同じ内容の問い合わせくり返し時間

TTL を 12 時間としているので、キャッシュが有効に機能しているなら同じネームサーバからの同じ内容の問い合わせは、12 時間経たないと重複しないと考えられる。しかし、図に示すとおり実際には 1 分未満の時間間隔で同じ内容の問い合わせが送られてくる場合がかなりある。この例では、100 秒未満の時間間隔で繰り返される問い合わせ数は、全体の 2 割を占めている³。このようなネームサーバはキャッシュ機構を実装していないか、キャッシュを用いないアプリケーションからの問い合わせを受けているか、あるいは問い合わせに対する答えに満足せず⁴同じ問い合わせを繰り返すものであると考えられる。

以上をまとめると、次のようになる。

- ・ TTL 値が極端に短い (5 分) と、全体の問い合わせ数、重複問い合わせ数共に極めて多くなる。
- ・ TTL 値がある程度 (1 時間) 長ければ、全体の問い合わせ数、重複問い合わせ数は TTL が極端に短い場合に比べて 1/10 程度まで減る。
- ・ さらに TTL 値を長くしても、全体の問い合わせ数や重複問い合わせ数に対して明確な効果はない。これは、一つにはキャッシュの指定とは関係のない問い合わせがあるためであると考えられる。

³なお、問い合わせパケットには ID が振られていて、ここでは異なる ID のパケットしか対象にしていないので再送ではない。

⁴例えば、NXDOMAIN(その様なドメイン又はホストはない)という回答を受けても問い合わせをやめないなど

4 問い合わせの性質と戦略的決定

前章の結果を踏まえ、全体の問い合わせ数、重複問い合わせ数を抑えながら TTL 値を短くする方法を検討する。

4.1 問い合わせの性質

実際に問い合わせ内容を調べてみると、いろいろなところから何度も問い合わせを受ける、いわば有名ホストが浮かび上がる。NTT の場合、情報サーバである `www.ntt.jp` が特異的に多く、各サブドメインのネームサーバ、メーリングリストサーバ、メールホストなどが続く。

図 5 は、TTL=12 時間における問い合わせ内容を、数の多い順にプロットしたものである。問い合わせを多くうけた上位 20 のレコードを抜き出してある。

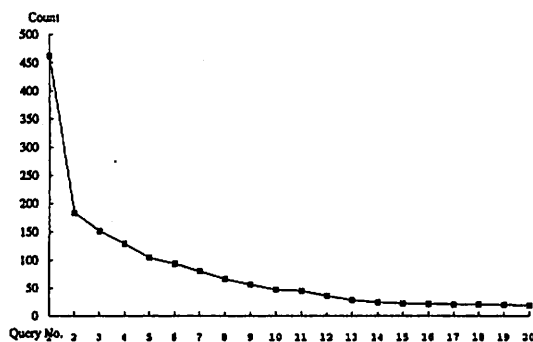


図 5: 1 日の問い合わせ上位 20 種の数 (TTL=12 時間)

問い合わせ内容は有名ホストに集中しており、上位 10 レコードに対する問い合わせ数の合計は、全体の 5 割近くになる。この傾向は、TTL を短くするにつれ顕著となり、TTL=1 時間のときには 6 割、TTL=5 分のときには 8 割近くの問い合わせが、少数のレコードに対するもので占められる。TTL の変化と、上位 10 レコードに対する問い合わせ数をプロットすると、図 6 のようになる。

図では、特に TTL=5 分の場合、上位レコードへの問い合わせ数が著しく増加していることがわかる。このことから、このような少数の特定レコードに対してのみ、長い TTL を与えてやれば全体の問い合わせ数削減に効果的であると推測できる。

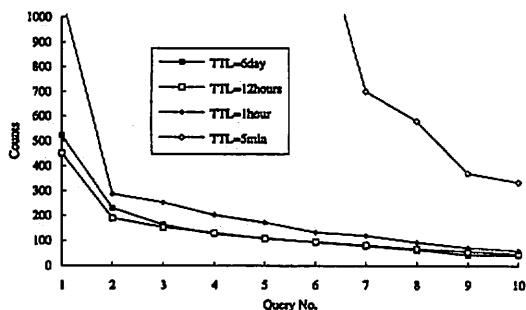


図 6: TTL が異なる場合の問い合わせパターン (上位 10 種まで)

4.2 手作業によるチューニング

実際に、デフォルトの TTL を 5 分にしたまま、外部から問い合わせの集中した 20 のレコードに対し、12 時間の TTL を与えて効果を見た。結果を表 2 に示す⁵。

表 2: TTL の組み合わせとその影響

TTL	5 分	12 時間	5 分+12 時間
総問い合わせ数	36615	2849	3443
全体の重複	87.0%	31.4%	33.8%

表に示したとおり、1100 余りのデータレコードのうちわずか 20 のレコードに長い (12 時間の) TTL を与えるだけで、サーバの負荷を十分抑えることができた。現在は default TTL を 1 時間とし、特定レコードに最大 12 時間の TTL を与えて運用している。

4.3 戦略的決定

前節にあげた手作業のチューニングによる効果を踏まえて、TTL の最適化を目的とした決定方法を検討した。

4.3.1 定常状態の把握

まず、制御の目的値を定めるために定常状態を把握する。観測値としては一日当たり総問い合わせ数、一日当たり重複問い合わせ率を採用する。例えば、NTT のネームサーバでデフォルト TTL=6 日とした場合、

⁵数値はいずれも一週間計測・一日平均

一日の総問い合わせ数は平均 2710 件、重複問い合わせの比率は同 28.6%となった。

4.3.2 管理モデルの決定

次に、ドメインとして受け入れる管理モデルを決定する。ここでいう管理モデルとは、以下に挙げる値の組み合わせで定義される、ドメインの管理方針であり、戦略である。

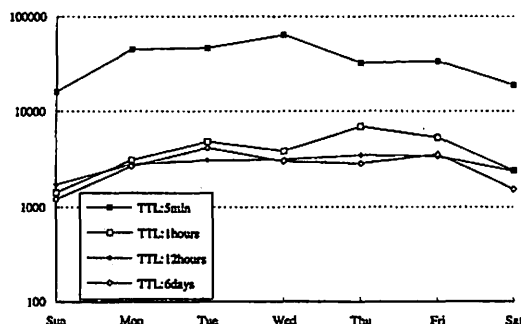


図 7: 総問い合わせ数の週間推移

1. 見直し間隔 (P)

動的な TTL 値の見直し間隔である。また、観測値の統計データを処理する時間間隔でもある。この間隔は、時間や曜日に伴う統計の揺らぎの影響を受けないよう長くとる必要がある。例として曜日に伴う総問い合わせ数の変化を、図 7 に挙げる。一日の時間的変化は、通常午前 9 時ころと午後 1 時ころをピークとする分布になる。

2. 上限総問い合わせ数 (Qmax)

1項の間隔における、外部ネームサーバからの問い合わせ数の上限であり、制御の目標値である。この値を越えないように制御をおこなう。

3. 上限重複問い合わせ比率 (Dmax)

1項の間隔における、外部ネームサーバからの問い合わせのうち同じ内容であったものが全体に占める比率である。この値を越える問い合わせ内容は、特に引き合いの強い物であると判断でき、長い TTL を与える候補となる。

4. デフォルト TTL (Td)

ネームサーバの SOA レコードに記述する minimum TTL で、デフォルトの TTL である。

5. 上限 TTL (Tmax)

制御に用いる最長の TTL 値である。

6. 固定的 TTL (Tfix)

制御の対象外とするレコードに与える TTL 値である。

7. 固定的レコード (Rfix)

固定的 TTL を与えるレコードで、ドメインにおいて重要であり、しかるべき手続きを伴わないと変更されないような管理の元にあるレコードを指定する。例えば NIC に登録したネームサーバのアドレスなどがこれにあたる。

例として、NTT ネームサーバの管理モデルを表 3 に挙げる。

表 3: NTT ネームサーバの管理モデル

見直し間隔	1 週間
上限総問い合わせ数	3000 件/日
上限重複問い合わせ比率	30 %/日
デフォルト TTL	1 時間
上限 TTL	1 日
固定的 TTL	1 日
固定的レコード	JPNIC 登録のプライマリ 及びセカンダリネームサーバ

4.3.3 決定フロー

前節で定義した管理モデルを用いて、図 8 のようなフローで TTL を決定する。

簡単にいうと、一定の時間内における全体の問い合わせ件数および重複問い合わせ比率をモニタしておき、これらが目標値を上回ったら問い合わせの多いレコードをコントロールリストに加えて長い TTL を与える。

TTL の変更によって直接影響を受けるのは重複問い合わせの比率であって件数ではないが、問い合わせの上位を占めるレコードに対して変更を適用するため全体の問い合わせ件数も減少させることができる。一方、図 5 に示したように上位以外の問い合わせ数は非常に少なく、およそ全体の 1%未満である為、制御しても全体に影響を与えることはできない。

5 終わりに

本稿では、運用中のネームサーバに対する外部ネームサーバからの問い合わせ内容から、従来経験的にきめられていた DNS のキャッシュ寿命の値を合理的に決定する方法を示した。この方法に従い、従来採用されることのなかった 1 時間という短い寿命値でも、デフォルトとして使用できることを示した。

残っている課題は二点ある。まず、本手法は NTT のネームサーバの統計データのみに基づいており、ドメインに独立かどうかは未検討である。特に小さなドメインでは問い合わせの種類が少ないので、ここで示したような手法がほとんど意味をなさないと思われる。逆にプロバイダの様に大きな組織で、他ドメインのセカンダリを多く引き受ける立場では、ここで管理モデルとしたドメイン当たり一日 3000 件の問い合わせというのは多すぎる値である可能性がある。

もう一つの課題は、極めて短い TTL 値 (5 分など) をデフォルトとして採用出来るかどうかの十分な検討がなされていないことである。TTL 値を極端に短くする実験は、セカンダリサーバに負担をかけるため、余り長期に行うことが出来なかった。

今後は、上記の課題に取り組むとともに、ここで用いた管理モデルを拡張して、ドメインの特性の測定や検知に用いることが出来るようにしたいと思っている。

最後に、本稿をまとめるにあたって実験に協力していただいた NTT COREnet DNS ワーキンググループの皆様には感謝いたします。

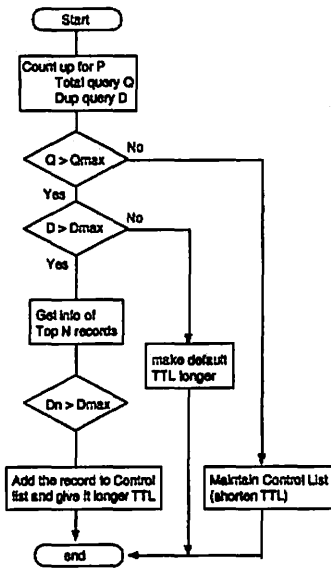


図 8: TTL 制御の基本フロー

コントロールリストに追加されたレコードについては P 時間後に TTL を見直す。個別のレコードに対して問い合わせ数と重複率を計算し、 Q_{max} の 10% もしくは D_{max} を上回っていたら 1step 長く、逆に Q_{max} の 1% を下回っていたら 1step 短くする⁶。TTL の変化ステップは図 9 に示すように長いほうには粗く、短いほうには細かくする。これは図 3 に示したようにある程度長い TTL については効果が緩慢であるためである。

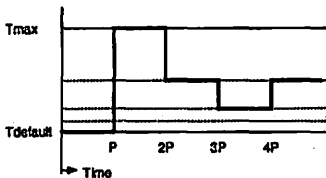


図 9: TTL 制御のステップ例

これまでに述べた決定の方法に従って、現在 ntt.jp のネームサーバを運用中である。サーバの負荷とネットワークトラフィックを抑えたまま、1 時間という短い default TTL が採用できている。

⁶ここでいう 10% とか 1% という値は実験に基づく経験値であり、ドメインによって異なる可能性がある。

参考文献

- [1] P. Mockapetris, *DOMAIN NAMES - CONCEPTS AND FACILITIES*, Internet Standard 13, available from <ftp://ds.internic.net/rfc>, 1987
- [2] C. Peckham, *domain-faq*, available from <ftp://ftp.njit.edu/pub/dns/Comp.protocols.tcp-ip.domains.FAQ>, 1995
- [3] P. Albitz, C. Liu, *DNS & BIND*, O'Reilly & Associates, 1992. (日本語版 浅羽・上水流訳、アスキー出版局、1995)