

電子メールの配信を高速化する SMTPfeed の設計と実装

中村 素典

京都大学 大学院 経済学研究科

京都市左京区吉田本町

motonori@econ.kyoto-u.ac.jp

インターネットでは、多数の人々にむけて一斉に同報通信を行うためのサービスの一つとしてメーリングリストが広く用いられている。メーリングリストとは、リストに登録されている受信者全員に対して同一内容の電子メールを送付するサービスであるが、もし受信者間において電子メールが到着する時間に大きな差があれば、メーリングリストを用いた議論に支障をきたすといった問題が生じる。そこで、本論文ではメーリングリストの配信処理における無用な遅延を抑える手法を提案する。この手法を SMTPfeed と呼ぶ sendmail の外部メーラとして実装し評価した。この結果、提案する手法が有効であることが示された。

Design and Implementation of the SMTPfeed to Improve Delivery Speed of Mailing Lists

NAKAMURA, Motonori

Graduate School of Economics, Kyoto University

Yoshida Honmachi, Sakyo, Kyoto

motonori@econ.kyoto-u.ac.jp

Mailing List is one of general services in the Internet for distributing copies of a message to a number of people. The copies of a message are sent as a e-mail to recipients in a list. If there are large differences in arrival time of the copies to each recipients, the delay hinders comfortable group communication. In this thesis, some methods to avoid unnecessary delays in delivery process of Mailing List is proposed. These methods are implemented in SMTPfeed, which is an external mailer for sendmail, and evaluated. The evaluation shows the proposed methods are effective.

1 はじめに

インターネットでは、1つのメッセージを多数の受信者に送るメーリングリストと呼ばれるサービスが広く利用されており、中には受信者数が1万人を越えるものも存在している。このようなメーリングリストの配信では必然的に受信者間におけるメッセージの到着時間に差ができてしまうが、一般に受信者の数が増えるほどこの差は大きくなる。しかし、スムーズなコミュニケーションのためにはメッセージの到着時間に大きな差があるのは望ましくなく、さらにサービスが有料である場合には、この時間差がサービス品質の差として大きな問題となりうる。したがって、受信者数の多いメーリングリストにおいて到着時間の差を小さくする、すなわち高速にメッセージを配信することは一つの課題である。

本論文では、受信者数の多いメーリングリストの配信を高速化するための手法を提案する。また、提案する手法を SMTPfeed として実装し、実際に運用に組み込んで評価を行った結果について示す。

2 既存の MTA の問題

メーリングリストの配信処理は MTA (Mail Transfer Agent) が受け持つ。MTA としてこれまで様々なソフトウェアが開発されてきているが、現在最も多く利用されている MTA の一つに sendmail*がある。sendmail は、古くから利用されているが、配信に時間がかかるという問題を持っている。この sendmail の配信の遅さは、並列に処理可能であるはずの DNS の問い合わせや SMTP¹⁾ による配信を逐次的に処理していることに起因している。

このような sendmail の問題を解決するため、過去に筆者は sendmail に高速化のための改良を試みた²⁾。この改良によって遅延は大幅に解

消したが、複数の SMTP を同時に処理していないため、SMTP 自体に時間がかかる宛先があると、それが後続する配信処理の遅れの原因となってしまうという問題があった。

一方、最近では sendmail より新しい MTA として qmail**や exim***, postfix****といったものが開発されている。これらの MTA は、開発当初からメーリングリストの配信を高速化するための工夫が施されている。しかし、例えば qmail による高速化はメールの配信を受信者ごとに完全に独立して行う方式を採っているため、DNS 問い合わせや SMTP の相乗り配信によるトラフィックが多くなりネットワークやサーバにかかる負荷が高くなってしまいう問題がある。また、exim は DNS への問い合わせが逐次処理のため、配送の開始までに時間がかかるという問題をもつ。そこで、これらの MTA の欠点を改善し冗長なトラフィックの削減と配信処理の高速化を試みる。

3 配信の高速化手法

本章では、メーリングリストの配信を高速化する上で必要となる手法を整理し、これらの手法を統合した新たな MTA の設計を試みる。

3.1 SMTP の並列処理

メーリングリストの配信のように1つのメッセージのコピーを多数のメールサーバに送付する形態のデータ転送では、同時に異なるメールサーバに対して SMTP コネクションを設定し並列かつ独立に処理することで、同時に設定可能なコネクション数まで並列度を上げることができ、配信時間を全体として短縮することが可能となる。

ただし、SMTP の仕様では1回のトランザクションにおいて指定できる受信者のアドレス数は最大で100までである。このため、1つのメールサーバに送るメールの受信者の数が100を越

*<http://www.sendmail.org/>

**<http://www.qmail.org/>

***<http://www.exim.org/>

****<http://www.postfix.org/>

える場合は複数回の SMTP トランザクションに分割する必要がある。このような場合、分割されたトランザクションは並列に処理可能ではあるが、送信先メールサーバの負荷の上昇を抑えるために逐次的に処理するものとする。

3.2 SMTP での相乗り配信

メールアドレスのドメイン部 (user@domain の domain の部分) が異なっても MX レコードで指示されているメールサーバが同じであれば、同一 SMTP トランザクションでメールを配信することができる (図 1)。このように、異なるドメイン部を持つアドレスについてもまとめて 1 つの SMTP トランザクションで配信してしまうことを配信の相乗り (piggybacking) と呼ぶ。

```
com1.co.jp IN MX 1 mail.isp.ne.jp
com2.co.jp IN MX 1 mail.isp.ne.jp
```

図 1: 同じメールサーバの異なるドメイン

相乗りの判定は、配信先となるすべてのアドレスについて MX レコードの検索をし、それらのレコードの第 1 位に同じホストが指示されているものを集めることによって行う。もし、そのホストへの接続が成功しなかった場合は、第 2 位以下を順に繰り上げて再び相乗りの判定を行う。これにより、あるホストが、あるドメインに対する第 1 位の MX になっており、かつ、別のドメインに対する第 2 位の MX になっている場合でも相乗り配信の余地が生まれ、トラフィックの削減につながる。

3.3 SMTP コネクションの再利用

1 つの SMTP コネクションは、複数の SMTP トランザクションの処理のために繰り返し利用することができる。コネクションの再接続のためのオーバーヘッドを避けるためにもコネクションの再利用は重要である。

あるドメインに対して MX レコードが複数存在する場合、MX レコードに定義されるプレファレンス値の小さい順で並べ替えてから利用し、もし、1 番目の MX レコードに指示される

ホストが応答しなかった場合、2 番目の MX レコードに指示されるホストへの配信を試みる。もしここで、2 番目の MX レコードに指示されるホストが、他のドメインの 1 番目の MX レコードに指示されるホストと同一であり、すでにこのホストに対して配信が完了していたとすると、そのために利用された SMTP コネクションをすぐに切断せずに再利用することで SMTP コネクションの再接続のためのオーバーヘッドを避けることができる。

そこで、SMTP コネクションの切断処理の直前で、まだ配信処理が完了していない宛先に対する MX のリストの中に切断しようとしているコネクションの接続先ホストと同一のものが残っている場合は、再利用の可能性に備えてコネクションをそのまま保持する。

もし、再利用のために残されたコネクションで飽和し新しいコネクションを確立することができなくなった場合は、古いコネクションから順に切断していくものとする。

さらに、宛先ホストへの到達性情報についても保持することで、配信不能なホストへの接続要求を何度も試みてしまうことを防止し、無駄な 75 秒の TCP 接続要求のタイムアウト待ち³⁾を避けることができる。

3.4 パイプライニング

SMTP では、クライアントがサーバへコマンドを送り、その応答が返されるのを待って次のコマンドを送る。しかし、直前のコマンドに対するレスポンスの内容に関わらず次のコマンドを先行して投入できる場合もある。このようなコマンドの先行投入によって、通信にかかる時間を積み込むことができ、SMTP トランザクション全体としてかかる時間を削減することができる。このようなコマンドの先行投入のことをパイプライニングと呼ぶ。ただし、サーバ側の実装がパイプライニングに対応しているかどうかをクライアントに伝える仕組みが必要であり、その方法は RFC2197⁴⁾に定義されている。

3.5 DNS の並列問い合わせ

メールアドレスに対応づけられるメールの送信先ホストのアドレスは DNS によって管理されている。メールの配信時には、メールアドレスのドメイン部毎に DNS に問い合わせを行い、送信先となるホストのアドレスを取得する。ここでネームサーバからの応答が得られない場合、メールの送信先ホストが特定できないため当該宛先に対するメールの配信を延期することになるが、ネームサーバからの応答がないことは、通常約 75 秒のタイムアウトによって判断される⁵⁾。

前述の相乗り判定するためには SMTP 処理の開始前に全ての MX が取得できていなければならないが、DNS の問い合わせを逐次的に処理した場合、このタイムアウト時間が積算されるため配信処理開始までの大きな遅延となる。しかし、DNS の問い合わせにおいてドメイン間の依存関係は存在しないため、問い合わせは並列的に行うことが可能である。DNS の問い合わせは UDP を用いてコネクションレスに行われるため、全ての問い合わせを並列的に処理することにより、理論的には高々約 75 秒で全ての DNS の検索を完了させることができる。ただし、実際には、一つのドメインに関する問い合わせにおいて、CNAME、MX、A の各レコードの検索を逐次的に処理するため⁶⁾、さらに時間がかかることがある。必要となる時間は CNAME レコードのチェーンの長さ依存するが、チェーンは非推奨であり⁷⁾、MX レコードの右辺に CNAME レコードによる名前を定義することも非推奨である⁶⁾ことから、一般的に最悪の場合でも 225(75×3) 秒程度に抑えられる (MX を指す CNAME が 1 段存在する場合)。

3.6 DNS 付加情報の利用

例えば MX レコードの検索のように結果がホスト名であるような問い合わせを行った場合、さらに続けて得られたホスト名に対する IP アドレス (A レコード) を問い合わせることが多い。このことを考慮し、MX レコードに関する問い

合わせへの応答には、付加情報として関連する A レコードが同時に返される*。しかし、この付加情報がクライアントで有効に利用されることは少ない。

特に大規模なメーリングリストの配信では、膨大な数の DNS への問い合わせが発生するため、問い合わせの回数を減らすことはクライアントとネームサーバの間のネットワークの負荷や、ネームサーバ自体の負荷を低く抑える上で有効である。付加情報を利用すれば、DNS の問い合わせ回数は場合によっては半分まで抑えられる (全てのドメインに対して、MX レコードが存在している場合)。

なお、付加情報として偽の情報を付加することによって偽のホストにアクセスさせることができるというセキュリティホールへの対策として手元のネームサーバを bind 8.1.1 以降のものにしておく必要がある。

3.7 DNS 情報の再利用

メーリングリストに、同一ドメインの異なるユーザが複数参加していることは少なくないが、このような場合、情報の再利用で同じドメイン部に対する問い合わせを何度も行うことを防ぐことができる。また、異なるドメイン部に対する MX が同じホストに向いている場合も情報を再利用することで、2 回目以降の問い合わせを省略することができる (付加情報として A レコードと一緒に返ってこなかった場合)。

さらに、応答が得られなかった問い合わせの記憶 (ネガティブキャッシュ) しておくことで、約 75 秒のタイムアウトを何度も待つことも防ぐことができる。

4 実装

前章に述べた手法を総合的に利用した MTA の能力を評価するため、メーリングリストの配信処理を担当する独立したプログラムとして SMTPfeed を作成した。SMTPfeed は sendmail

*ただし、応答を返すネームサーバがその MX レコードに関連する A レコードの情報を把握している場合のみに限られる。また、NS レコードに関する問い合わせなどでも付加情報が返される。

から起動される外部メーラとして動作するが、sendmail と SMTPfeed とのインタフェースには LMTP⁸⁾ を用いているため、LMTP をサポートしている MTA であれば sendmail 以外のものからでも利用可能である。また、再送処理やエラー通知を SMTPfeed の呼び出し側に任せてしまうことができるため、SMTPfeed 側の機能を配信処理のみに絞り込むことができる。

LMTP は、SMTP を基に拡張されたプロトコルであり SMTP と同様に 1 回のトランザクションで指定できる受信者の数の最大は 100 である。しかし、メールの配信の並列度を上げるためには、SMTPfeed にできるだけ多くの受信者アドレスを一度に受け渡す必要がある。そのため、呼び出し側から SMTPfeed に受信者アドレスを渡す場合にはこの上限を緩和しておく。sendmail に対しては上限を任意に設定できるようにパッチを作成した。

前章で示した DNS 情報の共有、相乗り判定、SMTP コネクションの再利用などといった様々な手法を容易に実現するために、SMTPfeed では子プロセスの生成は行わず、全ての処理を 1 つのプロセス行うように設計した。これにより、プロセスの生成を行わないため速度低下につながる余計なオーバーヘッドも発生しない。SMTPfeed は、イベント駆動型の一つのプロセスとして動作し、多数のソケットを同時に扱うことで DNS への複数の問い合わせや複数の SMTP トランザクションを同時に処理する。また、BIND で提供されている DNS の問い合わせを行うリゾルバは並列的な問い合わせには対応していないため*、同時に複数のドメインについて問い合わせる独自ルーチンを持たせ、応答に含まれる付

加情報も有効に利用できるようにした。

ディスク上のファイルアクセスは処理が遅くなる要因となるため極力削減することが望ましいが、メールのサイズは大きくても数 M バイト程度で最近の計算機の主記憶の容量に比べて十分小さいことや、SMTPfeed では再送管理を呼び出し側に任せており処理中の計算機のクラッシュからメールの紛失を守ることは呼び出し側の責任であることなどから、SMTPfeed では全ての情報を主記憶に保存しディスク上のファイルアクセスは一切行わないようにした。

SMTPfeed は 1998 年 12 月現在、バージョン 0.90 が公開されている。詳細については、<http://www.kyoto.wide.ad.jp/smtppfeed/> を参照して頂きたい。

5 性能評価

SMTPfeed の性能評価は、他の MTA との比較および単独での配信能力の 2 つを調査することで行った。

5.1 他の MTA との比較

sendmail, SMTPfeed, qmail, exim, postfix のそれぞれについて、ドメインの異なる 30 人の受信者を持つメーリングリストでの配信時間を測定した。インターネット環境でそれぞれ 5 回ずつ測定したものを表 1 に示す。なお、測定に用いたメールサーバは、Pentium/90、主記憶 32M の BSD/OS 3.0 である。なお、測定はすでに手元のネームサーバに必要なデータがキャッシュされている状態で行った。

表 1: 5 種類の MTA での速度比較

sendmail 8.9.1	34/38	39/43	39/44	39/44	34/38
smtppfeed 0.90	4/8	3/7	4/8	3/7	4/8
qmail 1.03	11/22	8/14	9/10	7/18	9/10
exim 2.10	4/8	4/9	4/8	4/9	4/7
postfix 19981230	5/9	5/10	5/9	8/9	5/9

(90%/100%配信時間)
(単位は秒)

*bind 8 の contrib には arlib という実装もあるが、MX には対応していない。

この結果から、sendmail が最も遅いことは確認できるが、その他の MTA については有意な差が得られておらずほぼ同程度の性能が得られている。有意な差を得るには、ドメインの異なる受信者の数を大幅に増やして測定する必要があるが、そのためには受信者数と等しい数のホストが必要であるため今後の課題としたい。

5.2 SMTPfeed の配信能力

SMTPfeed はすでいくつかのメーリングリスト・サーバで実際に利用されている。それらのサーバにおけるログの中から注目すべきものを抜粋したものを表 2 に示す。

表 2: 受信者数の 90% への配信時間

受信者数	時間	通/秒
253	14	18.1
1213	29	41.8
3747	44	85.2
5141	192	26.8

(単位は秒)

この結果から、最も高い性能が発揮されたときで、1 秒当たり 85 通程度の配送性能が得られていることがわかる。

6 おわりに

本論文では、メーリングリストの配信を高速化する手法について提案するとともに、その手法を SMTPfeed として実装し評価を行った。その結果、qmail, exim, postfix 等に劣らない性能が得られていることが確認できた。

今後の課題としては、多くの受信者を持つメーリングリストでの他の MTA との性能比較、SMTPfeed を sendmail 以外の MTA と組み合わせ使用した場合の評価、sendmail の内部処理の高速化などが挙げられる。

また、LMTP をインタフェースとしたことで、計算機が処理中にクラッシュした場合の 2 重配信の可能性が大きくなったり、配信先が DSN⁹⁾ に対応していない場合に配信完了通知が出せな

いという問題があるので、これらを解決する方法を検討したい。

謝辞

SMTPfeed の実装に関して多くの御助言を頂いた東京大学の下川英敏氏、デバッグと性能評価に大いに御協力頂いた大阪大学の松下誠氏、その他、アルファテストのバージョンから運用に利用して頂き様々なフィードバックをお送り頂いた多くの方々に感謝する。

参考文献

- [1] Postel, J. B.: *Simple Mail Transfer Protocol*, RFC821, Aug. 1982.
- [2] 中村素典: 「メーリングリストの配送の高速化」, 第 25 回 jus UNIX シンポジウム論文集, Sep. 1995, pp. 21-31.
- [3] Stevens, W. R.: *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley, Jan. 1994, p. 235.
- [4] Freed, N.: *SMTP Service Extension for Command Pipelining*, RFC2197, Sep. 1997.
- [5] Albitz, P., Liu, C.: *DNS and BIND, Second Edition*, O'Reilly & Associates Inc., Jan. 1997, pp. 98-101.
- [6] Partridge, C.: *Mail routing and the domain system*, RFC974, Jan. 1986.
- [7] Mockapetris, P.V.: *Domain names - concepts and facilities*, RFC1034, Nov. 1987.
- [8] Myers, J.: *Local Mail Transfer Protocol*, RFC2033, Oct. 1996.
- [9] Moore, K.: *SMTP Service Extension for Delivery Status Notifications*, RFC1891, Jan. 1996.