

マルチプロトコル対応メッセージゲートウェイの実装と評価

An Implementation and Evaluation of Multiprotocol Message Gateway

木下 雅文†
Masafumi Kinoshita

相楽 恭宏†
Takahiro Sagara

中原 雅彦†
Masahiko Nakahara

1. はじめに

携帯通信キャリアにおいて、メール、SMS、コンテンツの暗号鍵配信など様々なメッセージサービスをユーザに提供するために、メッセージゲートウェイの需要が高まっている。メッセージゲートウェイはキャリア設備網に設置され、SMTP や HTTP、携帯業界団体 OMA (Open Mobile Alliance) が策定した SMPP [1]や PAP [2]などのプロトコルを組み合わせて、サービスを実現している。メッセージゲートウェイは、様々なプロトコルの組み合わせやシステム構成に柔軟に対応し、早期のサービスインが可能で、数千ユーザの大規模システムとして運用できる性能が求められている。

本稿では、上記要件を満たすべく、構成定義ファイルの変更のみでカスタマイズ可能で、様々なメッセージ中継サービスへの適応を可能にするマルチプロトコル対応メッセージゲートウェイを提案する (以下、メッセージゲートウェイを AMG と呼ぶ)。また、実機において AMG のメッセージ中継性能の評価について議論する。

2. マルチプロトコル対応ゲートウェイの構成

AMG の構成を図 1 に示す。AMG は、様々なキャリアグレードのメッセージサービスへ適応するため、構成定義ファイルを変更するだけで入力と出力のプロトコルの組み合わせを自由に変えることができる機能を持つ。また、AMG はサービスの拡張などの機能追加も少ない開発工数で実装することができる。以下、AMG の構造について説明する。

2.1 マルチプロトコル対応フレームワーク

AMG は、SMTP、SMPP などメッセージプロトコルの共通基盤として適用することができる 2 つの特徴、セッション制御部とメッセージ制御部から構成する。セッション制御部は、入力 (受信) 処理を行うフロントセッション制御部と、出力 (送信) 処理を行うバックエンドセッション制御部から構成し、それぞれに SMTP、SMPP、PAP、HTTP などメッセージサービスで想定されるプロトコルを実装している。通常、プロトコルに準拠したサーバは、セッションの状態である “ステート” に応じて、プロトコルで定義された処理を行う。AMG のセッション制御部はこのステートの仕組みを様々なプロトコルに適用するために汎用化し、フレームワークとして実装している。AMG の各プロトコルは、フレームワーク上にプロトコル固有の処理ステートと、フレームワークが提供するコネクション制御などを行う基本通信処理ステートを組み合わせることにより、プロトコルに準拠したセッション処理を実現できる。たとえば、図 1 において、SMTP 受信処理ステートは、基本通信

処理ステートでアクセプト処理を行うステート “通信開始” と、SMTP 固有のステートでコマンド HELO の受信待ち状態の処理を行う “HELO 受信”、コマンド MAIL FROM の受信待ち状態の処理を行う “MAIL 受信” などのステートの一連の組み合わせにより、SMTP に準拠した処理を実現する。AMG はこのステートの組み合わせを構成定義ファイルに記述し、セッション制御部は構成定義ファイルに記述された順にステートを遷移させ、対応するステート処理関数を実行する。これにより、構成定義ファイルに記述された各プロトコルのステートの変更、追加により、各プロトコルの処理をカスタマイズすることができる。また、各プロトコルの実装はプロトコル固有の処理ステートだけを開発すればよいので、スクラッチ開発と比較し、開発工数を削減できる。

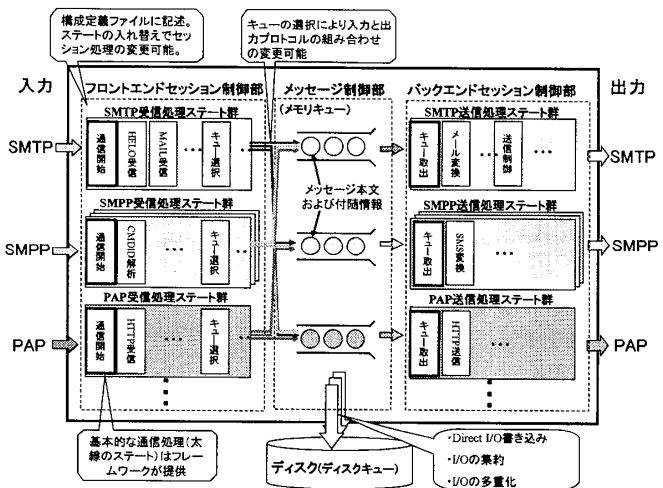


図 1 マルチプロトコル対応ゲートウェイの構成

一方、メッセージ制御部は、メッセージおよびメッセージキューの管理を行う。メッセージ制御部は、メモリキューとディスクキューから構成し、通常はメモリキュー上で処理を行う。ディスクキューは、ディスク上のキューのバックアップであり、障害等が発生しても信頼性のあるメッセージ配送を保証する。キューとバックエンドセッション制御部のステートは結びついており、フロントエンドセッションの “キュー選択” ステートを変更することにより、入出力のプロトコルの組み合わせを変更することができる。

2.2 メッセージ制御部の I/O 効率化

AMG は、メッセージ中継処理においてボトルネックであるメッセージ制御部のディスクへの書き込み処理を効率化する。以下、ディスク I/O インタフェースへの書き込み処理を効率化する 3 つの機能について述べる。

† (株) 日立製作所システム開発研究所
Systems Development Laboratory, Hitachi, Ltd.

(1) Direct I/O インタフェース書き込み機能

OSS (Open Source Software) の中継サーバの実装方式では、OS のファイル I/O インタフェースに対し読み書きを行う。しかしながら、AMG はディスクへのバックアップのため、すなわち I/O インタフェースに対し書き込み処理しか行わないため、ファイル I/O のキャッシュは不必要となる。そのため、AMG はファイルキャッシュを介さず、Direct I/O インタフェースへ書き込みを行い、これによりファイルキャッシュへのコピーとファイルキャッシュ処理の2つのオーバーヘッドを削減した。

(2) ディスク I/O 集約機能

OSS の中継サーバの実装方式では、1 メッセージ受信するごとに、ディスク I/O リクエストへ発行している。しかしながら、メッセージのサイズが小さいと I/O リクエストが多量に発生し、ディスクの I/O 性能限界に達する。そこで、AMG は複数のメッセージから発行されるディスク I/O リクエストを1つに集約して、I/O 数を削減した。

(3) ディスク I/O 多重化

OSS の中継サーバの実装方式では、書込みできるディスクは1つのみであった。AMG は、複数のディスク I/O インタフェースへリクエストを分散し、中継処理性能を向上させた。

3. 評価

本稿で提案した AMG のメッセージ中継性能を評価するために、MTA (Mail Transfer Agent) として動作する AMG の性能を測定した。また、比較のため広く MTA として使用されている sendmail, postfix も同時に測定した。

3.1 評価環境

評価環境の構成を図 2 に示す。測定対象である各アプリケーション AMG, sendmail, postfix がインストールされている PC サーバと、性能評価を行う評価プログラムの PC サーバの2台で構成する。評価用プログラムは、SMTP クライアントと SMTP サーバで構成し、SMTP クライアントは AMG へメールを送信し、SMTP サーバは AMG から中継されたメールを受信する。SMTP クライアントは 20~500msg/sec の送信レートで、メールを 10 分間送信し、正常応答を受信したメールの数を評価する。SMTP クライアントが送信するメールは、メールサイズ 400Byte が 70%、14Kbyte が 30% で構成する。上記 2 台の PC サーバは、SAS 73GB 10,000rpm RAID1 ディスクと、SATA250GB 7,200rpm RAID5 ディスク (2つのディスク I/O 性能はほぼ同等) とを備える。

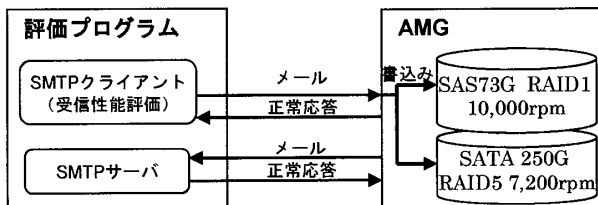


図 2 評価環境

3.2 結果と考察

各アプリケーションの限界性能と CPU 利用率の比較結果を図 3 に示す。AMG の 1 ディスクを使用したときの限界性能は、283msg/sec であり、2 ディスクを使用したときの限界性能は、498msg/sec であった。AMG の 2 ディスクを使用したときのメッセージ中継性能は、sendmail の 32 倍、Postfix の 6.2 倍であることを確認した。CPU 負荷は AMG が最大 27%、sendmail, postfix は 20% 以下しか使用していないため、各アプリケーションのボトルネックは CPU ではないことがわかる。

また、postfix の限界性能である 80msg/sec のメール中継時におけるディスク I/O リクエスト数の比較を表 1 に示す。postfix は 1160req/sec の I/O リクエストを発行しているのに対し、AMG は 1/10 以下の I/O リクエストしか発行していなかった。AMG は、I/O を集約することにより、I/O リクエスト数を削減していると考ええる。

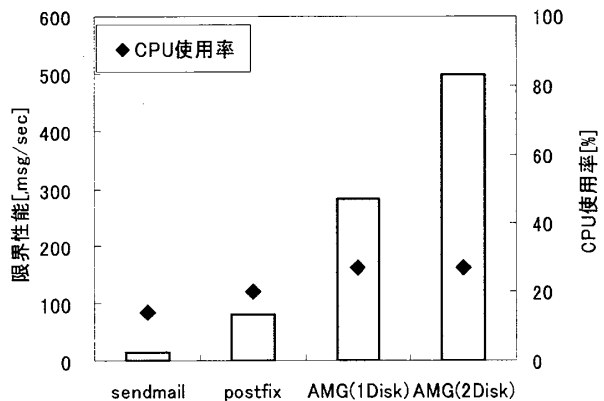


図 3 限界性能と CPU 使用率の比較

表 1 ディスク I/O リクエスト数の比較

	postfix	AMG(1Disk)	AMG(2Disk)
I/O リクエスト数 [req/sec]	1160	73	105

4. まとめ

本稿では、メッセージプロトコルの共通基盤として適用可能であるセッション制御部およびメッセージ制御部を持つマルチプロトコル対応フレームワークと、その上にプロトコル固有処理を実装するマルチプロトコル対応メッセージゲートウェイのアーキテクチャを提案し、様々なメッセージ中継サービスへの適応を可能にした。

また、メッセージ制御部のディスク I/O の効率化により、メッセージ中継の高速化を実現した。そして、MTA として動作する AMG を実機で性能評価し、postfix の 6 倍の性能である 498msg/sec のメッセージ中継性能を確認した。

参考文献

- [1] Open Mobile Alliance : Short Message Peer to Peer Protocol Specification version 5.0 (2003).
- [2] Open Mobile Alliance: Push Access Protocol Version 29-Apr-2001 (2001)