

分散 IMAP4 サーバー— NomaMail(2)

2 F - 5

実現方式

坂倉隆史 井上淳 清原良三 黒田正博

三菱電機(株)情報技術総合研究所

1 はじめに

分散IMAP4サーバー—NomaMail はインターネットを使用して、イントラネット内のメールサーバーに安価に、かつ、安全にアクセスする手段を提供するもので、ユーザーはどこにいてもイントラネットでのメールアカウントを使用してオフィスと同じメール操作環境で利用できる。

本発表では、現在試作評価中の NomaMail の実現方式について述べ、試作を通して明らかになった問題点を考察する。

2 システムモデル

システムモデルとしては IP リーチャブルなサブネット毎に IMAP4 レプリカサーバーを配置したものを想定した。実現は、その対象をインターネット接続されたイントラネット、及び、インターネットとする。

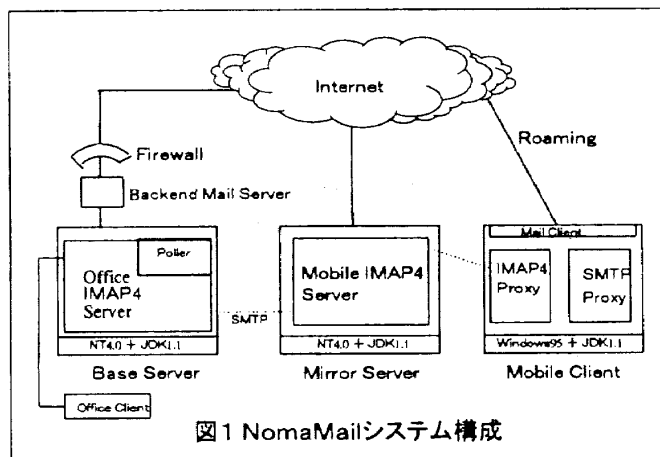
3 システム構成

分散IMAP4サーバーを実現するため、試作システムは図1のようにシステムを構成する。

Base Server はイントラネット内のメールサーバーが存在する LAN に接続されてにあるメールサーバーで、本システムは既存メールサーバーからメールデータを取り込むことにより動作する。本システムによる分散IMAP4サービスは既存のメールシステムに変更を加えることなく付加することが可能である。

Pollerと呼ぶ機能により、バックエンドメールサーバーから取り込んだメールデータを基にOffice IMAP4サーバーはファイアウォールの内側にあるメールクライアントに対してサービスを行う。メールデータはファイアウォールを越えて、インターネ

ットに専用線接続されたMirrorサーバー上で動作する Mobile IMAP4サーバーにデータ同期される。



Mobile IMAP4サーバーはOffice IMAP4サーバーと同様に同期されたデータを基にインターネットへローミング接続するメールクライアントに対してサービスを行う。

メールデータは暗号化の上各IMAP4サーバーに管理されているが、モバイルクライアント上でメールデータを取得するときは、モバイルクライアント上で動作するIMAP4プロキシ経由で行う。メールクライアントからIMAP4プロキシへのfetch命令を解釈して、メールデータを復号化の上メールクライアントに応答する。

モバイルクライアントからのメールの送信はSMTPプロキシを経由して行われる。メール送信先のアドレスをチェックしイントラネット内へのメールであったならばメールデータを暗号化して、本システムの該当するユーザアカウントに送信する。

4 JAVA および JAVA オブジェクトデータ同期機構の採用

本システムの開発言語として JAVA を採用した。理由はプラットフォーム非依存で動作環境を選ばぬこと、比較的短期間でプロトタイピング可能なことと、我々が現在開発中である MNCRS* 準拠の JAVA オブジェクトデータ同期機構をファイアウォールを越えてのメールデータ同期機構として利用可能であったためである。

5 JAVA オブジェクトデータ同期機構上のデータ構造

JAVA オブジェクトデータ同期機構が提供するオブジェクトコンテナ(以後 SyncStore と呼ぶ。)上に各ユーザーのメールデータを構築することにより実現している。

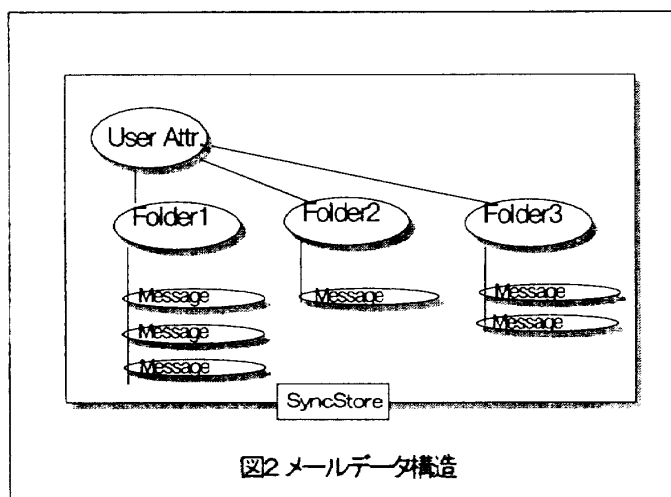


図2 メールデータ構造

図 2 に示すように1ユーザーに一つの SyncStore が割り当てられ、ユーザー属性情報を格納するオブジェクトをルートとして、メールフォルダ管理オブジェクト、メールメッセージオブジェクトをそれぞれ管理している。メッセージオブジェクトにはメールデータの他、IMAP4 プロトコルによるメッセージの既読、削除などの状態をあらわすフラグも格納する。

6 メールデータ同期とコンフリクト解決

システム構成で説明したように、SyncStore に格納されたメールデータを使用してイントラネット内とインタ

ネット接続サーバー上で IMAP4 サービスを行うが、サイト間のメールデータの同期にはメールメッセージの通過を許す構成のファイアウォールを越えてデータ同期可能な SMTP Synchronizer と呼ぶ、データ同期機能を利用して行う。JAVAオブジェクトデータ同期機構には利用するTCP/IPやSMTPといった通信方式に応じて Synchronizer と呼ばれるデータ同期機能を選択することができる。

SMTP Synchronizer による一回のデータ同期には比較的長い時間を要する。データ同期処理中も Office IMAP4 サーバー、Mobile IMAP4 サーバーへのユーザーアクセスは制限しないので、双方で同一ユーザーでメールクライアントからステートフラグ等の更新処理を行った場合データ更新コンフリクトが発生する可能性がある。データ同期機構は更新コンフリクトが発生した場合、アプリケーションが提供するリコンサイルメソッドにより、その解消を図る。現在メールデータコンフリクト発生時には、データ削除については残す方向で、各 IMAP4 ステートについてはメールクライアントにおいてのユーザーオペレーションが尊重される方向で、リコンサイルが行われる。

7 おわりに

現在 NomaMail システムの試用評価において、上記メールデータコンフリクトの他、Java VM 上で動作することに起因するシステムスケーラビリティ、応答性能が課題として挙がっている。アプリケーションによるチューニングの余地を検討しつつ、処理系での改善もスコープに入れ今後検討する。

参考文献

- [1] Network Working Group RFC2060 Internet Message Access Protocol - Version4rev1
- [2] Mobile Network Computer Reference Specification Version1.0: <http://www.mncrs.org>
- [3] 清原 良三、他「分散 IMAP 4 サーバー - NomaMail(1) 概要」本講演集 2 F04
- [4] 小野 良二、他「モバイル通信向け遠隔データ同期ミドルウェア JCapita」情報処理学会第56回全国大会講演論文集(3), pp. 534-535, May 1998

*MNCRS: Mobile Network Computer Reference Specification