

データベースを活用したWebメールシステムの開発

酒井 貴史¹・松本 慎平²・鶴沢 偉伸³・伊延 詩織²

¹電気電子情報工学専攻, ²制御情報工学科, ³都城工業高等専門学校 電気情報工学科

一般的なメールシステムを利用する場合、メールデータの分散という運用上の問題が発生する。この問題を解決する一つの手段として、Webメールシステムの利用が考えられる。しかし、一般的なWebメールシステムでは、通常のメールシステムに比べ運用規模が大きく、セキュリティやレスポンスに関する専門的な知識が要求されるため、利用者自身あるいは専門外の人員による恒常的な管理・運用は現実的に困難である。本研究では、メールデータの分散という問題をWebメールシステムで解決するため、DB(データベース)技術に注目した。DBを用いることにより、サーバ上のメールデータ自体に対して様々な加工や操作を加えることが可能となる。また、DBはデータ量の増加による急激な処理能力の低下を防ぐことができ、さらに不正アクセス等への対応が比較的容易であるといわれている。こうしたDBの特徴を踏まえた結果、本研究では、DBにより高いカスタマイズ性を維持しながらも利用者自身による容易な管理と利用者の利便性向上を実現できるのではないかと考え、DBを組み込んだWebメールシステムを設計・構築した。その結果、レスポンス向上やセキュリティ強化を実現しながらメール検索等の付加機能の実装を達成し、かつ利用者自身の手で管理運用可能な拡張性の高いシステムを構築した。

キーワード：データベース, Web, メールシステム

1. 緒言

現在、コミュニケーションツールの一つとして、電子メールが日常生活における様々な場面で用いられている。我々の身の回りにおいても、携帯電話をはじめとした情報通信技術は日を追うごとに普及しており、メールソフトの需要は今後も増加傾向にあることが予測されている。

メールソフトの需要増加とその普及に伴い、メールソフトの利用方法は、個人対個人のコミュニケーションツールに留まらず、利用者の目的に合わせて多様化する傾向にある。メールソフトの利用形態の多様化により引き起こされる様々な機能の要求に対応するため、近年では基本的な送受信機能以外の付加機能を実装した数多くのメールソフトが開発されている。具体的には、ユーザのアドレス管理やメール検索等の機能、メールマガジンやML(メーリングリスト)といった機能を実装したメールソフトなどをあげることができる¹⁾。

メールソフトに実装される付加機能の多くは、プログラミング言語におけるメール送受信APIの利用により、利用者自身の手で様々な機能を容易に実装させることが可能である。また、APIを利用することで、ユーザの利用形態に合った様々な改良をメールソフトに加えることが可能である。しかし、メールシステムの運用上の問題であるメールデータの分散という根本的な問題は、APIの利用や

改良だけでは解決することができない。

メールデータの分散という運用上の問題を解決する一つの手段として、Webメールシステムを挙げることができる。Webメールシステムを用いることにより、ユーザは任意の端末にあるWebブラウザを通して、メールデータを統括的に管理・操作できる。例えば、サラリーマンが出張先でメールの確認を行う場合、従来のメールシステムであれば、メールデータの入った利用者自身のPCを常に持ち歩かなければならない。一方、Webメールシステムを用いることにより、出張先の端末から社内宛のメールを確認することができる。Webメールシステムはメールデータを一元的に管理しているため、メールデータの分散を防ぐことが可能である。しかし、一般的なWebメールシステムは通常のメールシステムに比べ運用規模が大きいため、セキュリティやレスポンスに関する専門的な知識が要求される。したがって、利用者自身あるいは組織内における専門外の人員による管理・運用は現実的に困難である。

本研究では、高いカスタマイズ性を実現しながらメールデータの分散という問題点をWebメールシステムで解決するため、DB(データベース)技術に注目し、DBを組み込んだWebメールシステムを構築した。DBの検索性能はデータレコード件数に常に比例するため、データ量の増加による急激な処理能力の低下を防ぐことができる²⁾。また、不正アクセス等への対応が容易であると一般的にいわれ

ている。本研究では、DBを用いることによって、高いカスタマイズ性を備えながら利用者の利便性を高めることができるのではないかと考え、システム設計・構築を行った。その結果、レスポンス向上やセキュリティ強化を実現しながらメール検索、予定時刻送信といった付加機能の実装を実現し、かつ利用者自身の手で管理運用可能な拡張性の高いシステムの構築を達成した。なお、本システムを動作させるため、DBMS(データベースマネジメントシステム)としてPostgreSQL、サーバソフトウェアとしてApacheを利用し、DBと連携したWebサーバを構築した。また、システムの根幹であるメールアプリケーションはJava言語とそのメールAPIであるJavaMailを、システムのユーザインタフェース開発にはPHP言語をそれぞれ用いた。

2. メールシステム

(1) 一般的なメールシステム

現在利用されている一般的なメールシステムとして、クライアント-サーバ型のメールシステムが挙げられる。クライアント-サーバ型のメールシステムは、メールクライアントとメールサーバから構成されている。メールクライアントには、主に以下の役割がある。

- メールサーバからメールをダウンロード(受信)する。
- メールサーバへメールを送信する。
- 送受信したメールを保存、管理する。

また、メールサーバには主に以下の役割がある。

- クライアントから受け取ったメールを目的のサーバへ送信する。
- 自身のドメイン宛のメールを受け取る。

一般的なメールシステムでは、利用者はメールクライアントを利用することで、サーバへのメール送信や、サーバからのメール受信を行うことができる。

(2) 一般的なメールシステムの問題点

一般的なメールシステムでは、一般的にサーバ側のメールデータを取得した後、クライアント側にのみメールデータが保存されていく。この方式は現在最も一般的なメールプロトコルとして用いられているSMTP・POP(Simple Mail Transfer Protocol・Post Office Protocol)により行われる。SMTP・POP通信では、メールデータはクライアントのメール受信処理に依存しているため、ひとりの利用者が複数のクライアントを利用していた場合、メールデータが分散することになる。メールデータの分散は、大規模な組織においては、業務上の大きな問題となることがある。従来のシステムでメールの分散という問題に対応するためには、利用者は常に同じクライアントを使い続けるか、各クライアント利用形態に応じてメールデータの保存場所を考え

なければならない。

メールデータの分散という問題点を解決する方法としては、メールデータをサーバに残すプロトコルとして開発された技術であるIMAP(Internet Message Access Protocol)の利用が考えられる。IMAPは従来のメールシステムを基礎として発展させたものであるが、SMTP・POP通信と比較してサーバの負荷が大きいため、大規模な利用者を対象とした運用は大幅な設備投資をしない限りは困難である³⁾。したがって、IMAPは従来のメール通信を発展させたプロトコルにも関わらず、現在においても普及はそれほど進んでいない状況にある。

(3) Webメールシステム

Webメールシステムは、クライアント-サーバ型のメールシステムと並び代表的なメールシステムのひとつである。Webメールシステムは、メールクライアントとメールサーバ間にWebメールアプリケーションを組み合わせたものである。利用者とWebメールサーバ間ではHTTP通信が利用され、Webメールサーバとメールサーバ間ではメールプロトコル(SMTP,POP)通信が行われる。つまり、Webメールアプリケーションは、HTTPを用いた利用者インタフェースを実装したメールシステムであると理解できる。

Webメールシステムでは、メールデータはWebサーバ上に常に一元的に管理・保存されており、利用者はWebブラウザからメール機能を利用する形式を取っている。したがって、Webメールシステムを用いることにより、メールシステムの運用上の問題であるメールデータの分散を解決することが可能である。また、ネットワークを利用可能な環境に端末があり、Webブラウザさえ用意されていれば、利用者はOS(Operating System)や場所を問わず、統括的にメールを管理・閲覧することができる。こうした理由からも、Webメールシステムは利用者の利便性を十分に考慮した運用形態であると考えられる。しかし、Webメールシステムでは通常のメールシステムに比べ運用規模が大きく専門的な知識が要求されるため、個人規模での運用は困難であると考えられる。

(4) JavaMailを用いたメールアプリケーションの開発

現在、いくつかのプログラミング言語では、メール送受信のフレームワークが登場してきている。フレームワークとは、アプリケーションソフトを開発する際に頻繁に必要とされる汎用的な機能をまとめて提供し、アプリケーションの土台として機能するソフトウェアのことである。インターネットをはじめとした情報通信技術の普及により、フレームワークなどソフトウェア技術に関する情報共有が容易になったため、様々な機能を有したソフトウェアの開発が個人規模で盛んに行われるようになってきている。メールシステムの開発も、その例外ではない。

Java言語においては、メール送受信機能を実装したフレ

ームワークであるJavaMailと呼ばれるライブラリが用意されている。JavaMailとは、Java言語でメールアプリケーションを開発する際に必要な機能を集約したライブラリであり、利用者はJavaMailのAPIを利用することにより、メールプロトコルの技術的な側面を意識することなく、メールアプリケーションを容易に開発することができる。2008年9月現在、JavaMailはVersion1.4まで配布されており、メール送受信、添付ファイル操作機能、利用者認証機能等が実装されている⁴⁾。JavaMailはメール送受信APIとしては代表的なものであり、JavaMailを用いて数多くの興味深いメールシステムが開発されている。例えば、JavaMailとClassifier4jを用いた知的な電子メールフィルタの開発や⁵⁾、放置されている未読メールを携帯に自動転送するメールシステムの開発⁶⁾をあげることができる。

3. データベース技術

(1) DBの利点

DBは一般的なファイルでのデータ管理に比べて、データの管理法として優れている点がいくつかある。特に、本研究ではデータベースを利用するにあたって以下に示す二つの特徴に注目した。

まず、DBの検索性能は、データレコード件数に常に比例するという点である²⁾。一般的なデータ探索手法として、単純法や、KMP法、BM法などをあげることができる。これら検索手法の計算量は、レコード件数を n 、データ長を m とした場合、それぞれ $n \times m$ 、 αn 、 n/m となる。しかし、どの手法であっても計算量はデータのパターンに依存するため、単純法やBM法の最悪計算量は指数関数的に増加する。一方、DBはデータを細かく分類して管理するため、ほぼ αn の計算量が保障されている。Webメールシステムの場合、実用段階では一般的に多数の利用者が多数のメールデータを扱うため、膨大な量のデータを管理しなければならないことが予測される。したがって、本研究では、メールシステムに対してDBを用いることにより、データ量の増加による急激な処理能力の低下を防ぐことができるのではないかと考えた。

つぎに、DBのアクセス方法は単純かつ明快である点をあげることができる。DBに蓄積されたデータへのアクセスは、DBMS(データベースマネジメントシステム)を仲介することで実現される。ここで、DBMSはアクセスを利用者とパスワードによって管理しているため、利用者IDとパスワードを適切に管理していればDBへの不正アクセスは不可能である。また、DBサーバの記憶装置に直接アクセスされた場合でも、データはDBMSによって暗号化されているので、データを取得することはできない⁷⁾。また、DBのセキュリティは出入りが明確なため、不正アクセスへの対策を比較的容易に立てることが可能である。

(2) DBへのアクセス

DBを操作する場合、コマンドラインからDBMSを起動する方法と、アプリケーションからDBMSへ接続する方法の二通りをあげることができる。ただし、後者の場合DBへ接続する仕組みをアプリケーションが実装している必要がある。ここでは本研究で用いたスクリプト言語とJavaアプリケーションによるDBの操作方法について述べる。

まず、スクリプト言語によるDB操作について述べる。本研究では、DBからデータを取得し利用者に表示するためにスクリプト言語を用いている。ApacheなどWebアプリケーションには、利用者からのリクエストに基づいてDBへアクセスする仕組みが用意されており、スクリプト言語を利用することでDB操作を容易に実装できる。スクリプト言語の代表的なもののひとつに、PHP(Hypertext Preprocessor)をあげることができる。PHPは記述が簡素である反面、充実したライブラリと非常に多くの関数を備えているため、LAMP(Linux, Apache, PHP)と総称されるようにWebアプリケーションの開発現場で広く用いられている⁸⁾。また、広く用いられているため、セキュリティ対策などに関するドキュメントが充実している。以上背景を考慮し、本研究はインタフェース開発に対してPHP言語を採用した。

つぎに、Javaアプリケーションからのデータベース操作について説明する。本研究では、メールサーバからメールを取得しDBにメールデータを蓄積するため、Javaアプリケーションを用いている。また、JavaアプリケーションからDBアクセスを行うために、JDBC(Java DataBase Connectivity)を利用している。JDBCはDBアクセスのための機能を集約したライブラリであり、DBへの接続やSQL文の発行を行うAPIを用意している。JDBCを用いることで、JavaアプリケーションからのDB操作を容易に実装することができる。

4. システム設計

(1) 提案システムの概要

本研究で提案するシステムでは、利用者によるDB内のメール操作はWebアプリケーションによって、またDB内のメールデータとメールサーバ間のやりとりはJavaアプリケーションによってそれぞれ実装されている。WebアプリケーションとJavaアプリケーションはそれぞれ独立して稼動するため、利用者は間接的にメールサーバとのメール送受信が可能である。メールサーバとWebアプリケーションのやりとりはDBを介して行っている点やメールデータがDBによって統括的に管理・蓄積されている点は、本システムの大きな特徴である。全てのメールデータをいったんDBに蓄積させ、その後同期的に送信処理を行うことで、メールの送信時間の制御や指定日時送信が可能となる。また、全てのメールデータはDBで管理されているので、DBの特



図-1 DBのテーブル定義

徴を活かした高速なメール検索や、メールデータに対して様々な処理や編集を加えることができる。これらDBの特徴を活用することにより、高いカスタマイズ性を維持しながらも利用者自身による容易な管理と利用者の利便性向上を実現できるのではないかと考え、システムを設計した。

(2) システムの仕様

一般的なWebメールシステムでは、(1)利用者認証、(2)メール操作、(3)ログアウトの3つのセッションから構成されている⁹⁾。本研究ではマルチアカウントを実装しているため、(4)アカウントの選択が実装されている。

利用者自身による容易な管理と拡張性を示すため、本研究では、Webメールシステムに対して以下に示す3つの機能を実装させた。

- 指定日時送信機能
- マルチアカウント機能
- 本文・件名等からの検索

DBでデータを格納するためには、テーブルを定義しなければならない。本研究で取り扱うデータは、利用者データとメールデータの二つに大別される。本研究では、これらの関係性を明確にすることにより、リレーショナルデータモデルでテーブル定義を行った。まず、利用者とメールデータのE-R図を作成し、E-R図の関係性を持つようデー

タモデルを設計することにより、図-1に示すテーブル定義を得ることができた。

(3) アプリケーションの機能

DBと利用者間のインタフェースとして開発されるWebアプリケーションは、以下の4つの機能を実装している。

- 利用者(ブラウザ)からDBへのメール送信機能
- DBから利用者(ブラウザ)へメール表示機能
- DBからメールを検索
- アカウントの切り替え

DBとメールサーバ間のインタフェースには、Javaアプリケーションが利用される。Javaアプリケーションが持つ3つの機能を以下に示す。

- DBからメールサーバへの送信機能
- 送信時刻を利用者指定日時になるように制御する機能
- メールサーバからDBへのメール受信機能

Javaアプリケーションの主な役割は、メールプロトコルを用いたDBとメールサーバ間の通信である。ここではDB接続機能とメール通信機能を持たせる必要があるため、DBとの接続はJDBCを用い、メール通信機能の実装にはJavaMailを利用している。ここで、数多くの利用者がそれぞれJavaアプリケーションを実行しようとした場合、サー

表-1 システムの実行環境

メールアプリケーション(Java)	JRE 1.6.0 update4, Java EE 6, JavaMail 1.4
メールサーバ	Pmail ver. 1.46
wwwサーバ	Apache 2.2.8
Webインタフェース	PHP 4.4.8
DBMS	PostgreSQL ver. 8.2

バに多大な負荷がかかることが想定される。したがって、本研究では、利用者の送受信要求に応じてJavaアプリケーションを実行するのではなく、サーバ上で一つのJavaアプリケーションを定期的に動かす方法を採用した。また、定期的に実行する際、指定された送信日時のメールだけを送信するように条件を設定している。なお、指定日時送信は本機能の応用である。

(4) データベースのセキュリティ

本システムでは個人情報を取り扱っているため、セキュリティを考慮する必要がある。Webアプリケーションを開発する場合、個人情報保護の観点からも、利用者から入力されるデータは厳重に管理しなければならない。

セキュリティ問題の一つとして、Webアプリケーションの入力フォームにSQL文を入力することによってDBへのアクセスを行う攻撃がある。これは「SQLインジェクション」と呼ばれている。SQLインジェクションを防ぐための対策として、入力フォームにおいてSQL文を受け付けさせないようにしなければならない。

PHPでは、SQLインジェクションを事前に察知し入力データを修正する`escape_string()`という関数が実装されている。`escape_string()`は、文字列からSQLインジェクションとなり得る文字を別の文字へ自動的に変換する機能を持っている。本研究では、`escape_string()`を用いることにより、SQLインジェクションによる不正なDB操作への対策を講じている。

5. システムの実装とその評価

(1) 実行環境

開発したシステムの実行環境を表-1に示す。Java言語とPHPは、動作するコンピュータ上に実行環境があれば、OSの環境に依存することなく利用可能であるため、汎用性の高いシステム構成により動作させている。

本システムの稼働方法について説明する。本システムでは、DBMS、Webアプリケーション、Javaアプリケーションの3つをそれぞれ動作させる必要がある。まずDBMSとJavaアプリケーションをコマンドラインで実行しておく。つぎに、Webアプリケーションで用いるPHPファイルをwwwサーバによって指定されたディレクトリに保存することにより、本システムを利用することができる。

(2) システムの評価

各機能が適切に動作しているかどうかを確認するため、本研究では、市販のPCに本システムを実装させ、運用実験を行った。運用実験では、利用者認証機能、マルチアカウント機能、メール送受信機能、その他の付加機能であるアドレス帳機能、指定時刻送信機能、メール検索機能の動作をそれぞれ確認した。

これより、利用者認証機能から順に説明する。Webアプリケーションのログイン画面を図-2示す。各利用者は本ページにより認証を行い、正しく認証された場合、次のアカウント選択画面へ移動しWebメールシステムの操作を開始することができる。運用実験により、不正なパスワードの入力やSQLインジェクションによる不正な操作では認証されないことが確認された。

ログイン画面で正しく認証された場合、アカウント選択画面へ遷移する。これはマルチアカウント機能であり、各利用者はアカウントの選択・新規追加を行うことができる。本システムでは複数のアカウントを管理することができるため、本システムを利用するとき、各利用者は必ずひとつ以上のメールアカウントを登録しなければならない。メールアカウントは、新規作成メニューを選択することでいつでも追加可能である。すでにアカウントを登録している利用者は、既存のアカウントをひとつ選択することによりメール操作トップ画面へ遷移することができる。運用実験から、アカウントの新規追加、アカウント選択とメール操作トップ画面への遷移は正しく機能することを確認した。

アカウントを選択した後、各利用者はトップ画面に移動し、メール操作を開始できる。ここではメールの新規作成、メールの送受信、アドレス帳の機能を提供する。

まず、メールの新規作成画面について説明する(図-3)。利用者は、本文、件名等、必要な情報をテキストフィールド上に入力する。本システムでは、付加機能として、ファイルの添付や送信時刻設定を行うことができる。

つぎに、送受信メール機能について説明する。初期設定では、件名、本文といった最低限必要な情報だけが表示される。ヘッダ情報など詳細が必要な場合は、ボタンをクリックすることによりヘッダ情報を表示させることができる。なお、添付ファイルは、受信メールの確認の際クライアントにダウンロードすることができる。運用実験により、メールの送受信・メールの閲覧は問題なく行えることを確認した。



図-2 ログイン画面



図-3 システム構成図

最後に、付加機能について述べる。本システムでは、各利用者にアドレス帳管理機能を提供している。メールを作成する際アドレス帳から相手の名前を選択するだけでメールアドレスを入力することができる。指定日時送信機能では、メール送信画面で送信日時を指定できる。その他、本文、件名、差出人等といった検索条件を送受信ボックスで指定することにより、メールを検索することができる。運用実験により、それぞれの機能を問題なく利用できることを確認した。

6. 結言

本研究では、既存のWebメールシステムが抱える問題を解決するため、DBを利用したWebメールシステムを提案した。また、DBMSとしてPostgreSQL、サーバソフトウェアとしてApacheを用いたDBと連携したWebサーバを構築し、Java言語及びJava言語のメールAPIであるJavaMailを用いたメールアプリケーションとWebインタフェースをPHP言語でそれぞれ開発し、本システムの動作を確認した。

PHP言語を用いたユーザインタフェース開発では、携帯電話をはじめとしたウェブブラウザを搭載する携帯情報端末からのアクセスをWebメールシステムの新たな利用方法として理解した。本システムのユーザインタフェースはWebサーバ上で動作するため、クライアントの環境に依存せず、ネットワーク環境とブラウザさえあれば携帯情報端末からでも利用可能である。また、本システムでは、ア

クセス時に利用者のブラウザが出力する環境変数を利用してインタフェースを表示させているため、利用者の端末に応じて異なったインタフェースを表示させることが可能である。ただし、携帯情報端末のブラウザは機種依存等の問題が存在する^{10,11)}。本システムが用意した携帯端末のインタフェースは限定されているため、今後複数のインタフェースを用意する必要があると考えられる。

市販のPCに本システムを実装させた運用実験では、高レスポンス性、セキュリティ管理の容易さを理解し、利用者自身の手で管理運用が可能であることを示した。また、メール検索、予定時刻送信といった付加機能を本システムに実装させることにより、拡張性の高さを明らかにした。

本研究では開発者と数名の協力者による運用実験を行ったが、実運用を行っていないため、大規模な利用者を想定した場合の性能評価や、システムの利便性に関する客観的評価を行うことができなかった。したがって、今後は数百名規模での試験運用をローカルエリア限定などで行い、アンケートにより客観的評価を実施する予定である。

参考文献

- 1) 電子メールの話, <http://www.orth.or.jp/Isikai/content/mailsoft.html>.
- 2) DBT-3によるPostgreSQL 8.2beta1のデータベース規模の違いに関する測定結果, <http://ossipedia.ipa.go.jp/capacity/CS0612150223/>.
- 3) Postfix研究会, Postfixで作る実践メールサーバ, 秀和システム, 2003.
- 4) 木下信, Java Mail 完全解説, 秀和システム, 2001.
- 5) JavaMail と Classifier4j でインテリジェントな電子メールフィルタを作成する, <http://japan.internet.com/column/developer/20051005/27.html>
- 6) 放置されている未読メールを携帯に自動転送する, <http://codezine.jp/article/detail/130?p=1>
- 7) PostgreSQL 8.0.3 文書, <http://www.postgresql.jp/document/pg803doc/html/encryption-options.html>
- 8) LAMP/LAPP: フルオープンソースという魅力的な選択肢,
- 9) <http://japan.cnet.com/column/watch/story/0,2000055932,20053965,00.htm>
- 10) 公文善之, WWWに基づいたインターネットメールシステムの構築, 青山学院大学大学院国際政治経済学研究所 修士論文, 1998.
- 11) 若林諒, 携帯電話を用いたWebシステム, 学習成果論文, 2006.
- 12) 渡辺圭太, JavaMailを用いたメールプログラムの作成, 鳥取環境大学環境情報学部 プロジェクト研究報告, 2004.

(2008.9.26受付)