

PHP による Web アプリケーションサーバの構築

— コミュニケーションサービス —

須川 和明

目次

1. はじめに
2. Web アプリケーションサーバ
3. PHP と WWW サーバ
 - 3.1. PHP の概要
 - 3.2. WWW サーバと関連サーバ
4. コミュニケーションサービス
 - 4.1. 電子メール
 - 4.2. ネットニュース
 - 4.3. コラボレーション
 - 4.4. ユーザ情報の管理と保護
 - 4.5. ソースプログラムの構成
 - 4.6. スクリーンショット
5. おわりに

1 はじめに

WWW システムは本来、構造化された文書をネットワークを通して共有するシステムとして開発された。このようなあらかじめ用意された文書を要求に応じて配布する静的なシステムに対して、サーバ内で WWW システムとは別のプロセスを起動して、文書やデータの加工を始め様々な処理を行い、その結果を WWW サーバを使用して配布する動的なシステムとして、CGI などが利用されるようになった。WWW サーバと CGI プロセスとの間のプロセス間通信を利用したこのシステムには、プログラム開発の繁雑さや処理速度の点などで限界があった。

Web アプリケーションサーバは静的な文書配布システムに加えて、WWW サーバにサー

バ内外での種々のプログラムの実行を可能にするシステムを組み込んだものである。

現在のネットワーク + コンピュータ利用環境においては、クライアントとして使用する PC で、Web ブラウザはほぼ100%インストールされていることが期待できるアプリケーションのひとつである。インターネットおよびイントラネットを利用してサービスを提供する方法としては、WWW サーバを使用した Web アプリケーションサーバによるサービス提供が簡単かつ確実に、しかもコストの削減にもつながる方法であると考えられる。

Web アプリケーションサーバは通常、WWW サーバと ASP や JSP、あるいは、JavaServlet などを用いて構築される。ASP は主として Windows 環境で IIS との組合せでのみ可能なシステムである。一方の JSP や JavaServlet は Jakarta-Tomcat サーバとの連携で利用する。Java のプラットフォームに依存しないという利点は、クライアントサイドでは非常に強力な武器となるが、もともとプラットフォームの固定されたサーバサイドではそれほど意味は持たないと考えられる。

本稿の目的は、オープンソースとして公開されているプログラム群だけを用いて、大学のような教育研究機関において必要と考えられる種々のサービスを提供する Web アプリケーションサーバを構築することである。

2 Web アプリケーションサーバ

Web アプリケーションサーバが提供でき

るサービスとしてどのようなものが考えられるであろうか。大学のような教育研究機関に限ると次のようなサービスが考えられる。

■コミュニケーション系 コミュニケーション系のサービスは、電子メール、ネットニュース、BBS、メッセンジャー、チャットなどのコラボレーションツールなど、個人対個人、特定のグループ内の個人、あるいは、個人対不特定多数との間でメッセージの交換を可能にするサービスである。

■データベース系 データベース系のサービスとしては、各種データベースへのアクセスを始め、大学などの教育機関では不可欠の時間割や講義要項データベースの作成、出席やレポート提出、成績データなどの管理を統合的に行う教育支援サービスなどがある。

オープンソースの TeX を使用した文書作成・編集機能を持つ文書管理システムも有効である。Web 上での文書の配布に適した PDF 文書や PostScript 文書への変換機能や作成・編集機能も利用できる。オープンソースの Zlib ライブラリへのインターフェースを利用してデータの圧縮を行えば、サーバの記憶領域を有効利用することも可能である。データベース系のサービスでは、様々なデータベースを利活用するための Web ベースのフロントエンドを提供して、ユーザに統一的なデータベースアクセスを可能にする。

■データ解析系 統計解析、数値計算、表計算など、データ解析系のサービスは、オープンソースで公開されている S 言語風の汎用統計解析言語である R 言語への Web ベースのインターフェースを提供できる。

データ入力支援として、商用の表計算ソフトで作成されたデータとの変換ツールや、解析結果のグラフ作成やプレゼンテーション用のスライドの作成には、オープンソースで公開されている MagicPoint などへのインターフェースも有効である。

■マルチメディア系 マルチメディア系のサービスでは、画像の作成や編集加工を支援する。また、オープンソースで公開されている Ming ライブラリなどへのインターフェースを用意すれば、フラッシュの動的な作成を含めてフラッシュデータの作成・編集などにも利用できる。

オープンソースのストリーム系サーバである Darwin Streaming Server や IceCast などと連携して、動画や音声などのストリーム系データへのインターフェースを用意することも可能である。

■ネットワーク系 プロキシ、ファイル転送、認証など、ネットワーク系のサービスは、学内専用サービスへの代理接続や、ファイアウォールを越えて安全に利用できるファイル転送サービス、あるいは、他のサイトへの接続に必要なユーザ情報などを一括管理して、インターネットのどこからでも Web アプリケーションサーバを経由してサービスを受けることを可能にする。

一方 LAN 内の各種サーバやクライアント PC などをネットワークを通して一括管理するシステムや、SNMP エージェントへのインターフェースを利用して、LAN 内のネットワークのトラフィックやネットワーク機器の監視を Web ベースのインターフェースを通して行うことができれば、緊急の場合にはサイト外からのアクセスも可能な遠隔管理システムとして利用することができる。

■開発系 Web アプリケーションサーバ自身の開発を支援するための開発系のサービスは、Web アプリケーションサーバを開発するために作成された関数群を再利用し活用するシステムとして構築できる。

Perl, Ruby, Python などのオープンソースの各種スクリプト言語による CGI の開発、および、PHP や JSP, Java などによる各種の Web アプリケーションの開発を支援するシステムの構築も有効となる。

本稿では、コミュニケーション系のサービスについて、Webアプリケーションサーバを構築する。

3 PHPとWWWサーバ

3.1 PHPの概要

PHP (PHP: Hypertext Preprocessor の再帰的頭字語) は汎用スクリプト言語であるが、HTML 文書中にスクリプトを埋め込む形で記述できるという特性を持つため、Webアプリケーションの開発に適している。このような特性を持つ言語としては、Windows 環境の IIS サーバで使用されている ASP や、オープンソースのスクリプト言語である Perl を埋め込むために Apache サーバ用のモジュールとして開発された mod perl などがある。

PHP は Perl などのスクリプト言語と同様にシェルからも使用可能で、WWW サーバと連携して CGI としても利用できる。通常は、同じくオープンソースとして公開された WWW サーバである Apache のモジュールとして組み込まれて作動するため、CGI や JSP、JavaServlet と比べて高速の処理が可能になる。

PHP は Rasmus Lerdorf によって Perl で書かれた参照追跡スクリプトから始まり、後に Personal Home Page (PHP の古い名前) ツールとして公開される。その後、HTML のフォームからの入力をデコードするツール FI (Form Interpreter) が PHP に組み込まれ PHP/FI あるいは PHP2 と呼ばれるようになる。

PHP3 から開発は個人の手を離れ開発者のグループにより組織的に行われるようになる。現在のバージョンである PHP4 ではスクリプティングエンジンとして Zend (Zend Technologies) が組み込まれ、パフォーマンスが飛躍的に向上するとともにセッション管理機能が標準的にサポートされるなど言語仕様で大きな改善が図られている。

PHP 言語は C 言語と非常に良く似た構文

を持ち、文字列に対する処理や正規表現では Perl 言語との共通点も多い。標準的にも非常に豊富なライブラリ関数が用意されているが、必要に応じてオープンソースのライブラリ群を追加して機能を拡張することができる。

3.2 WWWサーバと関連サーバ

WWW サーバとしては、Netcraft の調査で世界の 6 割以上を占めるといわれるオープンソースの Apache サーバを使用する。Apache は現在、Apache 1.3.x と Apache 2.0.x のふたつのバージョンが並行して公開されメンテナンスされている。Apache は必要な拡張機能を DSO (Dynamic Shared Object) モジュールとしてサーバ起動時に組み込んで使用することが可能である。

PHP はこのような DSO モジュールとして組み込まれて使用される。このため CGI のように別のプロセスを起動する必要が無く、サーバの負荷も最小限に抑えることができるとともに処理速度の点でも飛躍的に向上する。

通信路を暗号化してパスワードを始めとする各種情報を保護するためには、オープンソースとして公開されている OpenSSL ライブラリと mod ssl を DSO モジュールとして組み込む。これによって WWW サーバは SSL 対応となる。

ユーザのアカウント情報などの管理には、データベースエンジンとして MySQL を使用する。MySQL は PHP の開発の当初から標準的に使用されるデータベースとして知られ、PHP と親和性の高いデータベースエンジンとして使用されて来た。MySQL はマルチユーザ・マルチスレッド対応の SQL データベースエンジンで、MySQL AB からオープンソースとして公開されメンテナンスされている。

電子メールをクライアントに配信するための POP サーバとしては、YAT サーバを使用する。YAT サーバは JEPRO (Japan Education system PROducts) からオープンソースとして

公開されている拡張版 POP サーバである。YAT サーバは標準的な POP サーバの機能に加えて、メッセージのフォルダへの分別機能などメッセージ管理機能を中心に機能が拡張されている。

ネットニュースサーバとしては、ISC (Internet Software Consortium) からオープンソースとして公開されている INN (Inter NetNews) を使用する。INN は当初 Rich Salz によって開発されその後 ISC にメンテナンスを委託された。INN は NNTP プロトコルを使用した Usenet System として広く利用されている。

4 コミュニケーションサービス

4.1 電子メール

WWW サーバを利用した電子メールサービスは、Web メールシステムという名前で、商用のプログラムやフリーのプログラムが多数開発され利用されている。標準的な WWW ブラウザの多くには、電子メールを始め、NetNews やメッセージサービス、チャットなどのクライアント機能が組み込まれている。このような状況で Web アプリケーションサーバに電子メールを始めとするコミュニケーションサービスを組み入れることの意義は、各種サービスを提供するサーバへのアクセスを Web アプリケーションサーバを経由 (代理) して行うプロキシ機能の活用によって、サービスの集中管理を図ることである。

電子メールの利用者は通常、クライアントの PC などのメーラ (MUA: Mail User Agent) を起動して、メールサーバに接続しメッセージをダウンロードする。メッセージはクライアントの PC に保存されサーバ上からは削除される。ユーザが利用するクライアントの PC がこれ一台だけであれば全く問題は無い。しかし、現在、大学や企業などのユーザの多くは、大学の実習室の PC、研究室の PC、事

務所の PC、自宅の PC、モバイル用のノート PC、友人宅の PC、携帯端末、携帯電話、インターネットカフェの PC などなど、多数の PC を利用せざるを得ない、あるいは、利用可能な環境にある。

このような環境であってもそれぞれの PC を適切に設定して使いこなせば、メッセージの分散を避けて必要なときに必要なメッセージを発信、および、受信することは可能である。しかし初心者の多くにとってはそれほど容易なことでは無い。

サーバからメッセージを削除しないような設定にして、すべてのメッセージをサーバ上に残しておくこともできるが、メールプールが肥大化してサーバを圧迫することになる。サイトのサーバの設定によっては一定量を越えると警告無しに削除されるか、メールを受信できなくなることもある。

一方、ユーザの多くは複数のサイトにアカウントを持ち、複数のメールアドレスを用途に応じて使い分けている。また、コミュニケーション系の別のサービスであるネットニュースやメッセンジャー、チャットなどは、それぞれ別のサーバ、あるいは別のサイトで運営されていることもある。PC で使用されている標準的なメーラや WWW ブラウザは、通常、複数のアカウントを管理する機能を持っているので、適切な設定さえ行っていれば、複数のアカウントを使い分けること自体は初心者でもそれほど困難なことでは無い。

問題は上記ふたつの状況が重なった場合、つまり複数のクライアント PC から、複数のサイトに分散した複数のサービスごとの多数のアカウントを適切に利用することができるかという点である。不可能とは言えないまでも容易なことでは無い。

Web アプリケーションサーバでのコミュニケーション系サービスの役割は、多数のサイトに分散して存在する複数のサービスとアカウントを一元管理し、自サイトも含め多数

のサイトへのサービスのプロキシ (代理) サーバとなることである。インターネット上のどこにあるクライアントからでも、WWWサーバに接続することさえ可能であれば、このWebアプリケーションサーバに接続することで、このサーバを起点に多くのサイトの多くのサービスへ、複数のアカウントを使い分けてサービスを受けることが可能になる。

サービスの結果であるメッセージなどの各種データも、このサーバ上で一元管理することでデータの再利用が容易になる。初心者にとって面倒なサイトごとのIPアドレスやポート番号、アカウント名やパスワードなどの情報もWebアプリケーションサーバ内で一元的に管理することができる。

ユーザにとって便利に見えるこのような一元管理されたシステムに欠点があるとすれば、このサーバがダウンするか接続不可能になったとき、全てのサービスから一度に締め出されてしまうことである。サーバのミラーリングなどによって、同一の環境を複数のサーバ間で自動的に同期を取りながら分散しておくことが必要になるであろう。

■メッセージの送信 電子メールの送信には、通常、サーバの所属するサイトのMTA (Mail Transfer Agent) を利用して、作成したメッセージをSMTPプロトコルで通信してメッセージの送信を委託する。

電子メールのメッセージは、ヘッダー部とひとつ以上のボディ部から構成され、ひとつ以上の空行で区切られている。送信先のメールアドレス (To:, Cc:, Bcc: など) や表題 (Subject:)などのヘッダー部、および、ボディ部のテキスト部分の入力にはHTMLのFormを使用する。

複数のボディ部を持つMIMEメールでは、添付ファイルなどをサーバへアップロードした後、データがバイナリデータの場合にはデータのエンコード (Base64, uuencode など) を行ってテキストデータに変換する。適当なバ

ウンダリ文字列を作成してボディ部の区切りとし、マルチパート部を作成する。

作成したメッセージはPHPのmail()関数を使用して送信する。デフォルトではWWWサーバと同じサーバのMTAに送られ、他サイトなどのMTAに転送される。

■メッセージの受信 サイトのMTAが他サイトのMTAから受け取ってサーバのメールスプールに保存しているメッセージを受信するには、通常、POPかIMAPプロトコルを使用する。POPはメールスプールに保存されているメッセージをクライアントに転送するだけであるが、IMAPはサーバ上にメッセージを残したまま、メッセージの仕分けなどの管理も可能にするプロトコルである。Webメールシステムでは、届いたメッセージは通常、サーバに残して管理するためIMAPが使われることが多い。

本学では、インターネット接続当初から、POPプロトコルを独自に拡張したYATサーバをPOPサーバとして使用している。主としてメッセージをサーバ側で管理する機能を中心に拡張されている。IMAPと異なる点は、メッセージをメールサーバ上で一括管理するのでは無く、ユーザのホームディレクトリ内に転送して管理することである。

YATサーバへのアクセスは通常のPOPサーバと同様、110番ポートに接続して行すが、メッセージを取得するとクライアントのMUA (通常はPCのメーラ) に送られると同時に、通常、ユーザのホームディレクトリ内のMailディレクトリ中にあるinboxという名前のディレクトリ (メッセージフォルダ) に、MH形式 (1メッセージ1ファイル) で保存される。この時点でメッセージはサーバのメールスプールからは削除される。クライアントのMUAでメッセージをサーバに残さない設定を行っていれば、ホームディレクトリに保存されたメッセージも削除される。Webメールシステムでは、メッセージをホー

ムディレクトリ内に残すことで (もちろん、不要なメッセージを削除することはいつでも可能であるが)、IMAP サーバと同様にメッセージを適当なメッセージフォルダに分別し、WWW サーバ上でメッセージの集中管理を行うことを可能にする。

YAT サーバへの接続には PHP の `fsocketopen()` 関数を使用する。接続が確立したら、`fputs()` 関数で YAT サーバへコマンドを送信し、`fgets()` 関数でサーバからのデータを受信する。

PHP には、POP プロトコルを利用するための関数群が用意されているが、本稿では YAT サーバの拡張コマンドも含めて独自の PHP 関数を作成し使用する。

■メッセージの管理 YAT サーバでは、POP サーバとしての機能に加えて、サーバ上でメッセージの管理を可能にする以下のようなコマンドが追加されている。

メッセージ操作

MCHK 新着メッセージ確認
 INC 新着メッセージ取り込み
 LIST2 マーク情報付のメッセージリスト
 XHDR ヘッダ情報取得
 MARK マークを設定
 UNMARK マークを解除
 STORE メッセージを登録
 UNDEL 削除のキャンセル
 RETREX 拡張データを得る
 SETEX 拡張データを設定
 MOVE メッセージ移動
 COPY メッセージコピー
 LIST3 マーク情報付のメッセージリスト

フォルダ操作

MBOX フォルダ機能チェック
 MBLST フォルダ一覧
 MBLST2 フォルダ ID, ニックネーム一覧
 MBCHG カレントフォルダ変更
 MBDEL フォルダ削除

MBRST フォルダ削除のリセット
 MBMAK 新規フォルダ作成
 MBREN フォルダ改名
 MBMOD フォルダアクセスモード指定
 MBNICK フォルダエイリアス情報
 MBNSET フォルダエイリアス情報設定

その他

DFREE ディスク残量検査
 GECOS ユーザの GECOS 情報取得
 WHO YAT server ユーザリスト取得
 CMDLST 有効なコマンドのリスト取得
 GET 情報取得
 PUT 情報設定
 PUTENV ユーザ環境変数設定
 GETENV ユーザ環境変数取得
 MBPUTENV フォルダ環境変数設定
 MBGETENV フォルダ環境変数取得

フォルダ操作機能を使用することにより、受信したメッセージを分別して保存するためのメッセージフォルダの新規作成、削除、改名を始め、フォルダ名の一覧表示、フォルダ内のメッセージの一覧などの操作が可能になる。この機能により、複数のサーバ上の複数のアカウントごとに、受信したメッセージを別のフォルダに仕分けして保存することも、一括して同一のフォルダに保存することもできる。

一旦、フォルダに保存されたメッセージは、メッセージ操作機能を使用して、メッセージを別のフォルダにコピーしたり移動したりすることも可能である。不要なメッセージは、ごみ箱フォルダに移動するか削除することになる。

このようなメッセージの管理は、ユーザがメッセージを選択しながら手動で行うことになるが、メッセージの振り分けルールをあらかじめ記述しておくことにより、To:, From:, Subject: などのヘッダ部やボディ部に含まれる文字列パターンにより、受信時に自動的

に振り分けることも可能である。

「その他」の機能を使用してユーザごとの環境設定をサーバ上で行うこともできるが、本稿では後述するようにユーザ情報はデータベースで一括管理することにする。

4.2 ネットニュース

ネットニュース (NetNews) は転送系の電子掲示板システムである。各テーマごとにニュースグループと呼ばれるカテゴリに分かれ、ひとつあるいは複数のニュースグループに、記事 (article) と呼ばれる意見を投稿し、不特定多数のユーザ間で意見の交換を行う。投稿できるユーザが限定されたニュースグループや、記事の査読を行うニュースグループもある。

投稿された記事は、階層的に構成されたニュースサーバ間を NNTP プロトコルを使用してインターネットを通して転送される。クライアントからサイトのニュースサーバに記事が投稿されると、ニュースサーバは記事をサーバのニュースプールに保存するとともに、階層的に構築されたニュースサーバ群の上流および下流のサーバ群に記事を転送する。受け取った上流および下流のサーバ群は、同様に記事を自サーバに保存するとともに、記事を発信したサーバを除く他のサーバ群に記事を転送する。

ニュースサーバが受け取った記事は、サーバのニュースプールにあるニュースグループ名ごとのディレクトリに1記事1ファイルの形式で記事番号のファイル名で保存され、設定された一定の期間を経た後に古いものから順に自動的に削除される。

ネットニュースの記事は、電子メールのメッセージとほぼ同様の形式で、空行で区切られたヘッダー部とボディ部からなる。ヘッダー部は記事が転送されて来たサーバ名を順に"! "で区切った Path: という名前のヘッダーから始まり、From:, Subject:, Date: などと続く。

電子メールとは異なり、宛先が To: から Newsgroups: に変わる。

■記事の送信 送信する記事の作成は、電子メールのメッセージの作成とほぼ同様である。To: の代わりに投稿するニュースグループ名を Newsgroups: に指定する。複数のニュースグループ名をコンマで区切って並べ、複数のニュースグループに投稿することもできる。Subject:, From: は電子メールと同様であるが、From: は省略することもできる。ヘッダー部の後に一行以上の空行を空けて、本文を含むボディ部を作成する。

作成した記事は、サイトのニュースサーバの119番ポートに接続し POST コマンドを使用して送信する。記事を受け取ったニュースサーバは、自身のサーバ名を Path: に、サイトの名称などを Organization: に記載したヘッダーなどを追加して、上流および下流のニュースサーバに転送する。

■記事の受信 ニュースプールに保存されている記事を読むには、サイトのニュースサーバに接続して、必要な記事を受信する。オープンソースとして公開されている代表的なニュースサーバである INN (InterNetNews) では、記事を送信するための POST コマンド以外に、以下のようなコマンドが利用できる。

INN のコマンド

- LIST ニュースグループの一覧
- LISTGROUP グループ内の記事一覧
- GROUP ニュースグループ指定
- BODY 記事のボディ部表示
- HEAD 記事のヘッダー部表示
- XHDR 複数記事のヘッダー表示
- NEXT 次の記事へ移動
- LAST 最後の記事へ移動
- QUIT 終了
- XGTITLE サブジェクト表示
- STAT 記事の情報表示

サイトのニュースサーバに届いているニュースグループの一覧および保存されている記事番号の範囲は、LIST コマンドで得られる。LISTGROUP コマンドでニュースグループを指定すると、現在保存されている記事番号のリストが得られるので、GROUP コマンドでニュースグループを指定した後、HEAD コマンドおよび BODY コマンドで記事番号を指定すると記事のヘッダー部とボディ部をそれぞれ受信する。

PHP には、POP、IMAP プロトコル同様、NNTP プロトコルを利用するための関数群が用意されているが、本稿では独自の PHP 関数を作成し使用する。

4.3 コラボレーション

チャットやメッセージングシステムなど、1対1あるいは複数のクライアント間で、リアルタイムにコミュニケーションを行うサービスを提供するシステムをコラボレーションシステムという。

電子メールもメーリングリストなどを構成して複数のメンバー間で議論を行うことは可能である。ネットニュース、掲示板も多数のユーザの間で議論を行うことを可能にするサービスである。しかし、いずれの場合もある程度のタイムラグが発生するため、緊急を要するコミュニケーションには不向きである。

本稿では、WWW でおなじみのチャットなどと同様のシステムを作成して、コラボレーションシステムを構築する。このシステムでは、コラボレーションに参加できるユーザを制限するために、あらかじめユーザ名を登録したコラボレーショングループを作成しておき、登録されたユーザ（グループのメンバー）間でのみコラボレーションを可能にする。

また、グループのメンバーが Web アプリケーションサーバに接続しているかどうかの情報も一定の時間間隔で収集し、必要であればコラボレーションへの参加を要請するメッ

セージを送ることも可能である。さらに、特別なグループとして、誰でも参加できるグループや、IP アドレスなどで制限した特定のサイト内や実習室内のクライアントだけに参加を許すグループなどを作成することも可能である。

4.4 ユーザ情報の管理と保護

電子メールなどのサービスを提供するためには、メールサーバのアカウント情報やアドレス帳などの情報を管理する必要がある。ネットニュースでは膨大な数のニュースグループの中から、日常的にチェックする購読中のニュースグループを選択して保存しておく。コラボレーションでは、会議に加わることが許されるメンバーを設定するコラボレーショングループの登録などが必要になる。その他、Web アプリケーションサーバとの関連で必要になる各種のデータも一括保存して一元的に管理する。

Web アプリケーションサーバを利用しようとするユーザは、サイトのログインアカウントで認証を受けた後、ユーザ名を含む Web アプリケーションサーバのアカウントをユーザ情報データベース上に作成する。データベースに登録する必要がある情報としては以下のようなものがある。

電子メール 利用するメールアドレスごとのアカウント情報（サーバ名、ユーザ名、パスワード、各種設定情報など）、アドレス帳、受信したメッセージのフォルダへの振り分けルールなど。

ネットニュース 購読中のニュースグループ名のリストなど。ユーザが投稿した記事のコピーなど。

コラボレーション コラボレーショングループに所属するメンバーのユーザ名のリスト、あるいは、IP アドレスのリストなど。このグループメンバーのリストは、電子メールシステムと連携してメーリングリストとしても

使用可能である。

その他 利用履歴情報、お気に入りやブックマークなどの URL 情報など。

ユーザ情報の管理には、オープンソースとして公開されているデータベースエンジンを使用する。このようなデータベースエンジンとしては、PostgreSQL を始め、mSQL や MySQL など多数あるが、本稿では PHP からデータベースエンジンへのインターフェースが標準で組み込まれている MySQL を使用することにする。

クライアント (Web ブラウザ) と Web アプリケーションサーバの間をネットワークを通して流れるパスワードなどの機密情報は、SSL で通信路を暗号化することでその保護を図る。

Web アプリケーションサーバへのログインの他に、他サイトのメールサーバへアクセスしてメールを取り込む場合にもパスワードの送信が必要になる。また、他サイトの SMTP サーバを利用してメールを送信する場合にもパスワードの送信が要求されることもある (POP before SMTP)。接続先サイトが対応していれば、APOP パスワードなどの暗号化したパスワードの利用も可能にする。

PHP による Web アプリケーションサーバでは、各セッションごとに必要な情報はパスワードの情報なども含めて、セッション変数に保存され次のセッションに渡されて行く。この過程でサイト内でのパスワードなどの情報の漏洩を防ぐために、必要なセッション変数の暗号化を行う。暗号化にはオープンソースで公開されている MCRYPT ライブラリを利用した PHP 関数を使用する。

4.5 ソースプログラムの構成

PHP ソースプログラムの構成を以下に示す。*.php は PHP プログラムファイル、*.inc は関数定義などのインクルードファイルである。

ログイン処理

login.php
logout.php

電子メール

mail login.php
mail list.php
mail body.php
mail header.php
mail create.php
mail send.php
mail mime.php
mail init.inc
webmail.inc
pop3.inc
yat5.inc

ネットニュース

news list.php
news send.php
news init.inc
nntp.inc
name.j

コラボレーション

col login.php
col list.php
col send.php
col init.inc
col.inc

ユーザ情報管理

db login.php
db access.php
db init.inc
mysql.inc

その他

javasert.inc

4.6 スクリーンショット

ログイン画面を図1に示す。ユーザはここでユーザ名とパスワードを入力して認証を受けなければ、次の画面に進むことができない。

図2は電子メールのメッセージ表示画面である。上部のフレームにメッセージのリストが表示され、メッセージのサブジェクト部分をクリックすると下部のフレームにメッセージ本文が表示される。添付ファイルがある場合は、本文の左側にファイルへのリンクが表示される。ブラウザで表示可能なデータであればクリックすると表示される。表示できないデータは保存用のダイアログが出て、クライアントのディスクなどに保存する。

図3は電子メールのメッセージ作成画面である。To:, Subject: などのヘッダー部とボディ部のメッセージはテキストボックスから入力する。下部に添付ファイル指定用のダイアログを出すボタンがある。

図4はネットニュースの記事選択画面である。ニュースグループを選択して、記事のタイトルのリストが表示されている。電子メールと同様にタイトル部分をクリックすると本文が表示される。

図5はコラボレーション画面である。会話に加わるグループ名を選択してメッセージを送信する。現在ログインしているグループのメンバー名も表示される。



図 1 : ログイン画面



図 2 : 電子メール表示



図 3 : 電子メール送信



図 4 : ネットニュース



図5：コラボレーション

5 おわりに

本稿ではオープンソースとして公開されているプログラム群だけを用いて、電子メール、ネットニュース、コラボレーションサービスなどのコミュニケーション系サービスを提供するWebアプリケーションサーバを構築した。また、コミュニケーション系サービスに必要な各種のユーザ情報や、サービスを利用して収集したメッセージなどの各種データをデータベースで一元的に管理する機能も用意した。

コミュニケーションサービスはネットワークを多用するサービスであるため、ネットワークを流れる情報を保護するための方策として、通信路をSSLで暗号化し、サーバに一時的に保存する必要のあるパスワードなどの情報も暗号化するなどの処置を取った。

しかし、ネットワークを通して一方的に送られて来る電子メールやネットニュースのメッセージには、ウイルスや情報の漏洩を図る様々なプログラムが仕掛けられていることがある。ネットワークを利用するサービスにおいては、このような危険な攻撃に対して常に対処しておくことが必要であろう。

ウイルスなどを防ぐためのシステムは、ク

ライアント側で行う商用のウイルスチェックプログラムなどが一般的であるが、サーバ側でチェックを行うゲートウェイシステムなども利用されている。オープンソースではないが、H+BEDVからはフリーのライセンスで利用できるメールチェック用のゲートウェイであるAntiVirシステムやウイルスチェック用のパターンファイルなども公開されている。このようなシステムをWebアプリケーションサーバと連携させることで、ウイルスチェックもサーバ側で一括して行うことが可能になり、より安全に電子メールを利用できる環境を構築することが可能になるであろう。

本稿では、コミュニケーションサービスについてのWebアプリケーションサーバを構築したが、第2節で述べたデータベース系やデータ解析系のサービスを始めとする、他の多くのサービスを提供するWebアプリケーションサーバの構築については、今後の課題としたい。

参考文献

1. B.Laure, P.Laurie, Apache ハンドブック, O'Reilly, 1999
2. L.Stein, D.MacEachern, Apache 拡張ガイド (上, 下), O'Reilly, 2000
3. S.Garfinkel, G.Spafford, Web セキュリティ&コマース, O'Reilly, 1998
4. J.Castagnetto et al., プロフェッショナル PHP プログラミング, インプレス, 2001
5. L.Atkinson, PHP プログラミング入門, ピアソンエデュケーション, 2000
6. 田中ナルミ, MySQL&PHP, ソフトバンク, 2001
7. R.J.Yarger et al., MySQL & mSQL, O'Reilly, 2000
8. C.Musciano, B.Kennedy, HTML, O'Reilly, 2000
9. S.Gundavaram, CGI プログラミング, O'Reilly, 1999
10. O.Kyas, インターネットセキュリティ, Thomson, 1997

11. R.Fielding et al., Hypertext Transfer Protocol-HTTP/1.1, RFC2616, 1999
12. T.Berners-Lee et al., Hypertext Transfer Protocol - HTTP/1.0, RFC1945, 1996
13. J.Klensin Ed., Simple Mail Transfer Protocol, RFC2821, 2001
14. J.Myers, M.Rose, Post Office Protocol-Version 3, RFC1939, 1996
15. R.Gellens et al., POP3 Extension Mechanism, RFC2449, 1998
16. M. Crispin, Internet Message Access Protocol-Version 4rev1, RFC3501, 2003
17. P.Resnick Ed., Internet Message Format, RFC2822, 2001
18. N.Freed, N.Borenstein, Multipurpose Internet Mail Extensions (MIME) Part One, RFC2045, 1996
19. N.Freed, N.Borenstein, Multipurpose Internet Mail Extensions (MIME) Part Two, RFC2046, 1996
20. K.Moore, Multipurpose Internet Mail Extensions (MIME) Part Three, RFC2047, 1996
21. N.Freed, N.Borenstein, J.Postel, Multipurpose Internet Mail Extensions (MIME) Part Four, RFC2048, 1996
22. N.Freed, N.Borenstein, Multipurpose Internet Mail Extensions (MIME) Part Five, RFC2049, 1996
23. B.Kantor, P.Lapsley, Network News Transfer Protocol, RFC977, 1986