

イントラネットにおけるLinux とJava を用いたWWW-DBMS 連携システムの構築

朝日放送株式会社

執筆者略歴

赤藤 倫久

1996 年 朝日放送株式会社入社
技術局放送運用センターにて TV マスタ業務を担当

1998 年 技術局システム部
現在，社内システムの運用・開発業務を担当

論文要旨

朝日放送では社内のインフラ整備もほぼ完了し，全社的に共有すべき情報をイントラネットで提供すべく電子化を進めている．その Web 技術の根幹を成す部分において Linux や Java が注目されているのは周知のことである．これらをどのような形態で導入するかは，組織によって状況は異なるものの何らかの検討をする時期にきている．

そこでこれら要素技術を理解，検討することを目的に総務部所轄の来客受付管理をシステム化した．具体的な構成は，PostgreSQL を導入した Linux サーバと WWW クライアントを Apache JServ を介した Applet / Servlet 通信で接続する 3 層モデルである．

また商用 WWW DBMS 連携システムが高価であることを踏まえ、従来より構築コストを格段に低く抑えることも試みた．今回，サーバプラットフォームの RedHat Linux 以外を無償で入手可能なツール群で構築し，機能性・信頼性の観点からの考察を行う．

論文目次

1 .	はじめに	3
2 .	来客管理の概要と電子化の経緯	3
3 .	システム構成	4
4 .	開発作業	
4.1	環境設定	5
4.2	コーディング作業	6
5 .	データの流れ	
5.1	利用者側の操作	6
5.2	データ処理	7
5.3	ユーザ認証	8
6.	考察	
6.1	Java 固有の性質	9
6.2	信頼性確保	10
7 .	おわりに	11
	参考文献	12

図表一覧

図 1	クライアント側 WWW 入力画面
図 2	システムの概念図

1. はじめに

朝日放送では、基幹業務と位置付けられる 1998 年 11 月に完成した営放システムを中心として社内システムが構成されている。それを軸に業務支援的なサブシステムが複数存在する。また社内の事務業務の電子化が進んでおり、蓄積されたデータはネットワーク資源として利用可能になりつつある。個別のユーザ環境については、社内 LAN 環境整備と PC 配布により電子メールや WWW は浸透している。対ユーザを考える上でいずれの場合も重要な問題となるのが共有情報の提供方法である。

インターネット標準のドキュメントシステムである WWW は、専用の端末を設けないことが大きな利点である。業務システムを WWW により提供すれば、これまでのようなインストールの手間が大幅に軽減される。その上、複数のシステムの統合環境への発展性を勘案すれば有力な選択肢となり得る。

このようなことからイントラネットは既に一般的な実用手法となり、その要素技術は日々進歩を続けている。中でも特に話題に挙がるのは、サーバとしての Linux や Java によるアプリケーション開発である。

最近の技術変化に追随するため、総務部業務である来客管理受付のシステム化にあたっては WWW DBMS 連携を用いて構築した。

構成は Linux サーバにデータベースおよび Web サーバを準備し、クライアント側のアプリレットとサーバ側のサーブレットが通信を行う。そのサーブレットがデータベースとデータの入出力を行う 3 層モデルである。

また企業が Linux に期待している理由には低コストという点も大きい。そこでさらに無償提供されているソフトを用いてサーバを構成することを試みる。このような無償ソフトと呼ばれるものの中には Apache や GNU プロジェクトがリリースしているソフトのように信頼性も高く、シェアの大部分を占めるものさえある。無償ツールの機能に足りない要素があっても、運用で補完することは可能であると思われる。単なるコストの削減で完結することなく、業務使用のための要件を満たすべく障害時対応などを検討する。

2. 来客管理の概要と電子化の経緯

当社では防犯上の理由から来客者にお客様バッチを発行している。お客様バッチは、当日来社したときに受付で名前・所属を記入し担当者に確認をとることにより発行される。当日までに来客予定が分かっている場合は、前もって所定の手続きを行うことで来訪時に来客者が受付票に記入することを省略することにしている。時間帯によっては受付が混雑することから、来客者の事前の登録を推奨している。

しかし総務部に出向き、手続きをすることになっているため事前登録の制度は活用されていないのが現状である。そこでイントラネットでサービスを提供して、自分の PC から入力することが可能となれば来客者の事前登録の利用率も向上し、スムーズな受付業務に貢献するのではないかと考える。

3. システム構成

商用 WWW DBMS 連携システムは DB 本体と接続ライセンスなどが、必要であることを考慮すると非常に高価となる。しかし設計者には予算の制約以上の機能、信頼性を期待されることも少なくない。そこで無償で提供されているツール群を活用し、且つ必要十分な性能を確保することを試みる。

商用の業務パッケージは使用しないため、設計・環境設定からコーディングまでの全てを行う。このことは EUC の促進や単なる技術取得だけではなく、今後の外部業者に発注する際の開発工程・工数の見積もり評価の参考にもなる。

Linux をサーバとして選択する理由は、低コストであることに加えて厳密なプロセス、メモリ管理により OS 自身が安定していることと、Windows NT より貧弱な CPU やメモリ環境でも快適に動作することにある。

システム環境はサーバ OS である RedHat Linux 5.2 , http サーバとして Apache , Java 仮想マシンを搭載したサーバソフト (いわゆる Servlet Engine) として Apache JServ Project により配布されている Apache JServ をそれぞれ用いることにする*1。この Apache JServ は Apache にアドインして稼働させる。

重要な部分である来客者データを蓄積するデータベースは PostgreSQL を選択した。

PostgreSQL はカルフォルニア大学バークレー校で開発され、現在は世界中のボランティアによって開発・運用されているオープンソース系では最も評判の高い DB である。商用 DBMS に求められるようなトランザクション機能*2やオンライン・バックアップ機能を提供している上、ORDBMS (Object Relational DataBase Management System) 機能*3を備えていることでも有名である。またバージョン 6.5 より排他制御がこれまでのテーブル単位から行単位となりトランザクション機能が大幅に向上した。このようなことから企業においてフリーのデータベース導入を考えるなら最有力候補となり得る^[1]。

PostgreSQL には JDBC ドライバや ODBC ドライバが添付されており、これを使ってサプレットからデータベースに JDBC により接続する。

開発環境は Sun によって配布されている JDK (Java Development Kit) , JavaSoft により配布されている JSDK (Java Servlet Development Kit) に含まれている Servlet API を用いた*4。

また総務部が利用する管理用機能は、データ管理・印刷などが必要となるためユーザーと分けて別アプリとして実現する。こちらは不特定多数ではなく使用端末が限定されることから Visual Basic を用いて開発した。この管理者用モジュールからは ODBC により直接データベースに接続する。

*1 <http://www.apache.org>

*2 関連する複数処理をまとめて単一の処理単位として管理する機能。

*3 オブジェクト指向技術を取り入れた DB 基本型のほかに自由に定義できるオブジェクト型が導入されている。データ型の継承も可能。

*4 <http://java.sun.com> , <http://www.javasoft.com>。正確には Linux の JDK は blackdown がライセンスを受けて移植をしている , <http://www.blackdown.org>

システム構成を下記に示す。

サーバハード	Intel Celeron 433 MHz , メモリ 128Mbyte いわゆるサーバ専用機ではなく通常の DOS/V 機
サーバ OS	RedHat Linux 5.2
開発環境	JDK1.1.7 , JSDK2.0 (Java Servlet API) 管理者用モジュールのみ Visual Basic 5.0
実行環境	http サーバ Apache 1.3.6 Servlet Engine Apache JServ 1.0
データベース環境	PostgreSQL 6.5.1
クライアント環境	Internet Exploere 3.x 以上 , Netscape 3 以上
ネットワーク環境	本社 基幹系 100Mbps 支線系 10Mbps 本支社間は 0.5Mbps ATM で結ばれ営放システムとその他で 0.25Mbps づつに分割して利用

4 開発作業

4.1 環境設定

サーバ環境構築の最初の作業は Linux のインストールである。最近では書籍などで情報を得られるため決して困難な作業ではない^{*5}。インストール中に問題になることがあるとすればデバイスドライバであろう。必要不可欠なデバイスとしては NIC^{*6} (Network Interface Card) が挙げられるが、やはりドライバが標準でサポートされているものを選ぶ必要がある。

市場に多く出回っているカードならばドライバがサポートされていることが多いが、特に Melco や Planex^{*7}のように Linux や Free BSD の対応情報を積極的にインターネット等で公開しているメーカーもある。ドライバが提供されている NIC ならば容易にインストールを進めることができるので、動作確認のとれているカードを前もって選んでおくことをお勧めする。

Linux のインストールが終われば、Web サーバとサーブレット環境など本システム固有の準備に入る。その Web サーバを稼働させる初期設定の際に Java 仮想マシン (Java VM) と Servlet API , Servlet Engine へのパスの記述が必要となるので JDK と JSDK はその前にインストールしておく。

JDK は VM 用バイトコード^{*8}の実行環境を含むシンプルなコマンドラインベースの開発環境である。GUI 設計では不便であるものの、複雑な画面設計を行うのであれば問題とはならないであろう。UNIX の motif^{*9}開発環境と大差はないと考えてもらえればよい。Web サーバの Apache には DSO (Dynamic Shared Library) 機能を用いて、サーブレッ

^{*5} ディストリビューションに対応する各種書籍が出版されているが、日経 Linux など雑誌も参考になる。

^{*6} Lan ボードや Lan カードがこれにあたる。

^{*7} <http://www.melcoinc.co.jp> , <http://www.planex.co.jp>

^{*8} 擬似的なコンパイル処理で生成される中間コード。Java VM により実行する。

^{*9} 商用 UNIX で採用されている GUI システムの一つ。

ト用のモジュールを Shared Library として組み込んで使用する。そのため通常のように configure するだけではコンパイルオプションが足りないので mod_jserv.c を組み込む追加の設定が必要となる。上記と Apache JServ のインストールでは、初期設定のままではコンパイルがうまく進まないところがあり、Makefile を必要に応じて書き換える作業が必要となる。インストールが完了後、コンフィグファイルを設定して Apache が起動すれば、Web サーバとサーブレット環境は完成である。

PostgreSQL はソースを展開しコンパイル、初期設定と JDBC ドライバのインストールをすれば準備完了である。

クライアント側はアプレットが実行できるブラウザさえあればよいが、種類やバージョンによってプラグインなどの対応が必要である。

4.2 コーディング作業

Java の開発作業において特記すべき点はサーブレット処理部分と文字コード処理である。サーブレットとの通信部分については http の通信が容易に行える URL クラス^{*10}を利用した。サーブレット開発では、データ文字列や SQL の書式を整えるためのデバッグは通信結果を随時ログに出力しての作業となった。

文字コードは Java では Unicode で取り扱うが PostgreSQL では EUC^{*11}である。このことを考慮に入れ JDBC を経由する際に文字化けが起こらないように適切にプログラム内で変換処理を行う必要がある。

Visual Basic においてもバージョン 5 から文字コードを Unicode で取り扱うことは、Java と同じである。しかし ODBC ドライバの日本語化が実現されておりコード変換の煩わしさはない。さらに ODBC はサーバとの通信部分をカプセル化しており、開発者はデータベース接続について主に SQL のみを意識すればよい。

Java の開発期間は小規模で簡便なシステムであったこともあり、約 10 日程度で大部分の機能が完成した。Visual Basic による管理端末の開発には 4、5 日程度費やした。

5. 処理の流れ

5.1 利用者側の操作

来訪予定のある社員は来客者の氏名、所属の情報を前もって Web 画面に入力する。その際に入力した情報は、入力した個人と管理者以外には見えない様にするため、パスワードによる認証管理を行う。これに対し管理者は、あらゆるデータ操作が行う必要があり、こちらは VB で作成した専用端末を用意することで対応する。

社員はデータ入力に際し、専用のダイアログに対してログイン操作を行う。認証が終わればメイン画面に必要事項の入力を行い、登録するだけでデータベースに登録される仕組みである。登録されたデータは Web 画面上のリストに当日以降分のみが表示され、随時その内容の更新・削除を行うことができる。クライアント入力画面を図 1 に示す。

さらに受付担当者に、登録された来客情報を提供する必要がある。当面は管理用アプリ

*10 WWW 上の資源にアクセスする規約である URL を取り扱うクラス。

*11 文字コード体系の EUC

から翌日分の情報を帳票として印刷して受付担当者に渡すこととした．このため運用上，
 入力締め切りが発生する．いずれは受付に PC を設置し，リアルタイムの最新情報を閲覧，
 利用できるようにする予定である．

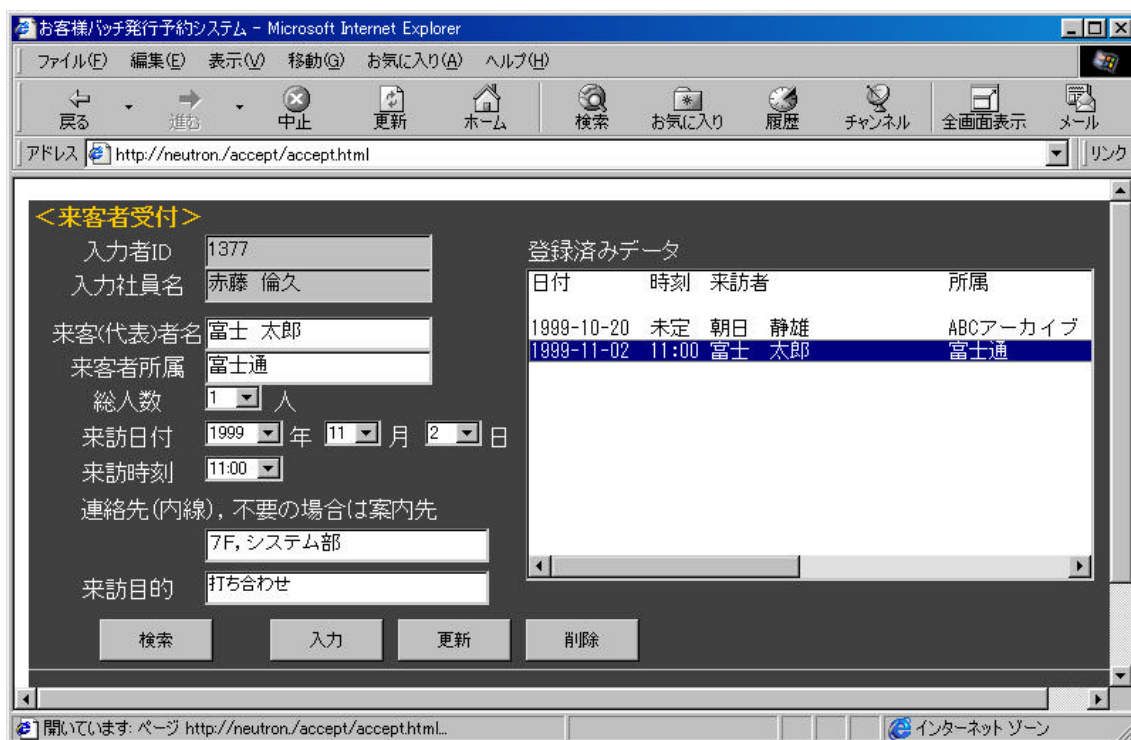


図1 クライアント側入力画面

5.2 データ処理

クライアント側のアプレットで入力されたパスワードやデータは，http を利用してサーバ側のサーブレットに送信される．サーブレットから Java プログラムとのインタフェースである JDBC を介して PostgreSQL に対しユーザ認証や必要情報の入出力を行う．アプレットに処理結果を送信し，受信したアプレット側にデータが表示される．つまりアプレットとサーブレット，DB から成る 3 層モデルとなっている．今回はサーブレットとデータベースを同じサーバに配置したが，別々のサーバに分けることも当然可能である．

アプレットからサーブレットにデータ文字列送信

サーブレットで受信したデータを元にデータベースに SQL を発行

検索登録結果をデータ文字列に変換

サーブレットからアプレットにデータ文字列送信

アプレットで受信データを処理して結果を画面表示

一方，VB で作成した管理用端末からは，ODBC を介して，PostgreSQL に対して直接入出力を行う．この端末からは来客データだけではなく，認証データの管理などすべてのデータ管理や帳票印刷を行うことができる．データベースへの接続手段をサーブレットからの JDBC と VB アプリからの ODBC のように複数の方法を準備することは，障害発生

時の原因の切り分けに役立つ。

管理用端末 (VB アプリケーション) からデータベースに SQL を発行

データベースから検索登録結果が受信

管理用端末で処理結果を画面表示

必要に応じて印刷

ここでシステム全体の概念図を図 2 に示す。

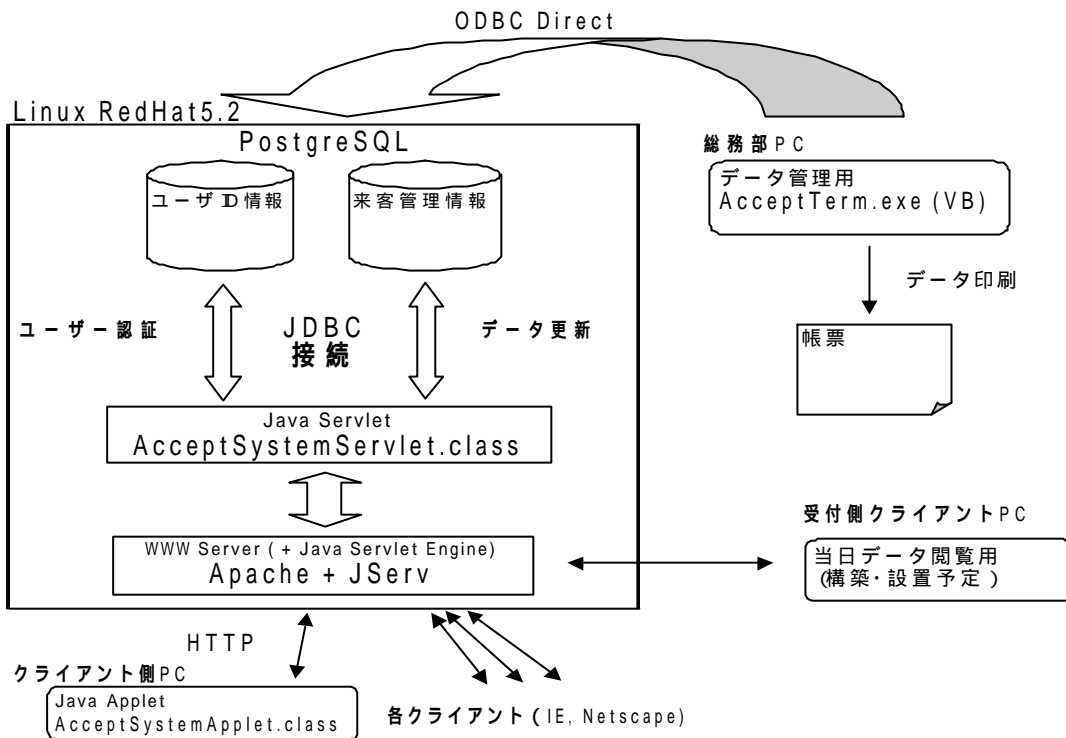


図 2 システムの概要

5.3 ユーザ認証

認証に使用するテーブルにはユーザ ID、氏名、パスワードおよび所属部署が収められている。社員の異動の際にはテーブル内の部署名の変更を行わなければならない。

管理端末からユーザを追加し、随時その内容を個別変更する機能は実装してある。さらに異動時期や機構改革の際の大量変更に備え、サーバ側には上位システムから取得したデータファイル进行处理してデータを一括して入れ替えるプログラムを C 言語で記述し用意した。ユーザデータの総入れ替えは、システム部の管理者のみが運用することを想定しているので、コンソールから対話形式で実行する簡便なものとした。

6. 考察

本運用に入り 2 か月が経過したが、利用法に対する問い合わせ以外は大きな混乱もなく、利用者は着実に増えているようである。特にシステムがダウンするような事象は発生していない。今後は利用者の希望や問題点を考慮して改修を行う予定である。これまでの考察を次に述べる。

6.1 Java 固有の性質

アプレットは html より細かな処理が行える半面、ダウンロードの時間が必要であることが問題点とされている。しかし当社内のネットワーク環境において、この時間は 5～10 秒程度と全く問題とされない長さである。処理時間は Pentium 133MHz クラスのクライアント PC からでも、データ 1 件の登録から結果通知まで本社内の場合で約 1 秒、東京支社からでも 2 秒程度と快適に使用するのに、十分なレスポンスを確保できた。

アプレットのもう一つの問題はブラウザの違いで見た目や挙動が異なる可能性があることである。バージョンによっては対応していない機能もあり、これらを根本的に解決するためには、ランタイムである Java プラグインの使用しかない。今回は Internet Explorer 3.x など古いブラウザに対応するためプラグインを使用した .html ファイルに必要情報を記述すれば自動インストールが可能だが、ブラウザのセキュリティ設定によってはそれを解除したり、各自直接インストールする手間が必要となる。

このプラグインのインストールや来客者管理システムの使用法はイントラネット上のホームページに載せてある。マニュアル類はユーザ教育の一環として準備すべきだろう。

サブレットは CGI とはプロセス管理の点で違いがあり、CGI が要求毎にプロセスを起動するのに対して、サーバの常駐プロセスがスレッドとして処理する。つまり初回のリクエスト以降、プロセスの初期化処理は必要ない。このためサーバの資源管理の点でサブレットは優れている。レスポンスについても CGI+Perl とほぼ同等の性能を実現できるようになったと報告されている^[2]。

またアプレットとサーバ間の通信の場合、Java のセキュリティ上の問題からダウンロードするサーバの名前解決^{*12}が必要とされている。これは Java 1.1 のセキュリティ仕様が、アプレットのダウンロード元のサーバ以外との通信が出来ないことに由来する。特にプロキシサーバやファイアウォールが置かれている環境では、名前解決がされないと接続やレスポンスに影響を及ぼす場合がある^[3]。当社ではプロキシサーバを経由するため Wins と hosts ファイルにより解決した。社内 LAN に接続しているマシンの 9 割以上が Windows マシンであり、これまで DNS ではなく Wins を利用してきたことが大きな理由である。hosts ファイルに関しては、業務の性質上の理由からシステム部以外で管理している特定グループ内のマシンを対象に利用した。

さらに現用のプロキシソフトの squid^{*13}をプロキシ側で名前解決を処理できる DeleGate^{*14}に置き換えることを考えている。これによれば Windows マシン以外からのアクセス時の名前解決にも利用でき、Java のマルチプラットフォーム性を活かすことができ

*12 TCP/IP においてホスト名と IP アドレスを結びつけること。

*13 最も著名なプロキシサーバソフト <http://squid.nlanr.net>

*14 電総研で開発されたプロキシであり DNS 名前解決、文字コード変換機能を提供 <http://wall.etl.go.jp>

と思われる。

ブラウザの種類の混在により通常版 html とプラグイン版 html が別となった。利用者の混同を避けるため、スタートページを設け、そこで Java Script によりブラウザの種類とバージョンを判別しそれぞれに自動で移動するようにした。

組織内に複数種類のブラウザが存在する場合には、機能や外観の統一と利用者の利便を図る工夫が必要不可欠であろう。

6.2 信頼性確保

商用ソフトはベンダーによるサポートが大きな安心感となっている。それに匹敵する材料を準備しなければ、無償の環境を業務システムとして導入することは難しい。まずその性能・性質の把握や他社の導入事例を調査することが重要である。

Linux は低コストと安定性に加え、リモートメンテナンスが可能であることを理由に導入する事例が大幅に増えているようである。問題点はハードのドライバ環境が NT に比べると未整備であることと、簡易 OS ではないのでセキュリティ対策をはじめ、管理者に相応の技術力が求められることである。

Java については、雑誌のサーバ側の実行環境におけるパフォーマンス評価が参考になるだろう。サーバとクライアントモデルの通信処理をサーバ内でシミュレートするローカルパフォーマンステストと、実際に 10Mbps の回線で通信させるネットワーク・スケーラビリティ・テストの調査結果が発表されている^[4]。これによれば同時接続数が非常に多い場合の永続的な安定性には、改善の余地があると報告されている。Linux の JDK 1.1.7 についてはその限界が 1000 程度のものである。当社では PC の総数が 800 台程であり、このシステムの場合の同時接続数は、おそらくこの 10%程度になると思われる。このようなことから現時点では、十分利用に耐えうる性能であると考えられるが、今後の Java プラットフォームの改良・発展にも期待したい。

PostgreSQL はオープンソース系で最も有力な DBMS である。先に脚注に示したホームページ上で多くの事例が紹介されているのをはじめ、利用者は着実に増えつつある。特に Apache との PHP/FI^{*15}を使った WWW DB 連携の事例は、よく紹介されている^{[5][6]}。また性能は Wisconsin Benchmark という付属ツールを使用して、ある程度の目安を得ることもできる。雑誌に掲載されている性能評価^[7]を参考にすると、100 万件を大きく上回る大量データ使用時の性能低下と、非同期モードでのデータの一貫性がないことが問題点として指摘されている。しかし同時 128 ユーザ内で同期モードによる 100 万件以下のデータのロードに際しては、実用範囲であると考えられる。

運用面についてはフリーソフトを利用する場合だけに限らず、バックアップ体制を充実させることが必要である。データを短い周期で採取しその版数管理を行えばバックアップとしては十分であろう。PostgreSQL には、データベースのテーブル内のデータ内容をファイルに保存する方法、テーブル単位でスキーマ定義やユーザ定義関数を含めてバックアップをする方法が提供されている。

さらにメンテナンスのためにシステムを停止することが許されるなら、サーバ側のパー

*15 HTML に記述したスクリプトをサーバ側で実行する WWW/DB 連携のミドルウェア

ジョーンアップやデータベースのメンテナンスの保守体制も取れるのではないだろうか。この保守の時間をとることができるかということも環境選定の重要な鍵となる。

ハードのバックアップ体制としては専用サーバ機ではない安価な通常の DOS/V 機を本番系と予備系用として2台用意した。低コストとライセンスフリーである利点を生かせば予備機を準備しておき、問題発生時にはハード全体を交換すれば、非常に早い復旧が可能となる。サーバ専用機の価格が通常機の5倍程度であることを考えると、より低コストで迅速且つ簡単な復旧体制を提供できると考えている。

また商用ソフトのように付属ドキュメントが充実していない場合もあるので、各自で必要な情報は入手する。幸い GNU をはじめとする有名なフリーソフトにおいては、インターネットなど^{*16}で豊富に情報提供や技術論が展開されている。PostgreSQL ではここ最近に 6.5 のリリースから 6.5.3 までの3回のバグ修正リリースが行われるなど、ユーザ活動も活発である。さらにシステム管理者グループの心得として、仕様マニュアル等を作成し情報共有を徹底することも忘れてはならない。

自組織の技術力や保守能力が求められることは、以上のような状況を考慮すれば決して困難な制約条件ではない。合理的な設計と運用上の工夫により、安定性・信頼性確保と障害時の速やかな対応は可能である。

7. おわりに

今回構築した来客管理システムは、企業内の WWW DBMS 連携システムの将来性を示すものである。通常クライアント端末側にある機能を、WWW 上で提供したことは緊急のモジュール変更を容易にし、運用開始前後の工数を減少させた。さらに他のシステムの汎用プロトタイプとしても応用することは可能である。

また低コストや優れた性能を理由に企業は Linux 等に大きな期待を寄せている。その点では信頼あるフリーソフトと運用上の配慮により構築費用を低く抑えることができた。しかし商用サポートがないことは、システム管理者に消極的な判断を導くようである。

オープンソースの世界では、自分で問題を解決する姿勢が重要である。それを補うようにフリーの世界ではネット上で活発な情報交換がなされている。能動的に参加すれば、商用サポートと遜色ない有意義な情報も得られることができるだろう。あとは制約条件を満たすように、プログラミング設計および運用管理を行えばよい。さらにライセンスフリーであることは、パッケージとして配布することや複製も可能となり二次利用にも期待できる。

EUC の促進という観点からサーバ/クライアントともに、Java でコーディングした本システムには大いに意義がある。Java 技術には今後のシステム構築に参考となる点が多いからである。先進技術の導入は利用者や管理者の新たな要望に応えられる可能性を与えてくれる。今回得た実際的知識は今後の社内システムに十分活用できるものである。基幹システムに対しての導入には解決すべき問題や制約があるものの、部分的応用は期待できる

^{*16} 例えば Java であれば <http://java-house.etl.go.jp/ml> , PostgreSQL であれば <http://www.sra.co.jp/people/t-ishii/PostgreSQL/index.html> が挙げられる。大抵の問題に対して大いに参考になる。

のではないだろうか。

試行的なものを含めた多種多様な実績を蓄積し、技術論議の中で利用者に還元できる方向性を見出したいと考える。

参考文献

- [1]「オープンソース DBMS の実力を探る」,日経 Linux 通巻 1 号,(1999.10) pp.104-127
- [2]原田洋子: Java Servlet 最新サーバプログラミング,集和システム,
1999 年 2 月 24 日
- [3]菊田秀明: 実践 JDBC, オーム社開発局, 1999 年 5 月 21 日
- [4]スコット・ブラモンドン: “パフォーマンス & スケーラビリティ”, 月刊 JAVA WORLD,
通巻 26 号,(2000.1), pp.32-37
- [5]高橋信頼: “Linux いよいよ企業システムへ”, 日経オープンシステム, 通巻 73 号,
(1999.4), pp.110-111
- [6]“活用”, 日経 Linux, 通巻 1 号,(1999.10), pp.96-99
- [7]高橋信頼: “検証 PostgreSQL の実力”, 日経オープンシステム, 通巻 80 号,(1999.11),
pp.175-189