
UNIX サーバから Windows 及び Linux 混在環境 への移行における効果的なシステム構築事例

株式会社 山下設計

■ 執筆者 Profile ■



廣瀬 由紀

1997年 (株) 山下設計入社
情報システム部所属
情報システム業務担当
1998年 構造設計部所属
情報システム業務担当



丸谷 周平

1993年 (株) 山下設計入社
構造設計部所属
プログラム開発業務担当
構造設計業務担当



山縣 信一郎

1995年 (株) 山下設計入社
構造設計部所属
構造設計業務担当
1999年 監理・コスト部所属
工事監理業務担当
2001年 構造設計部所属
構造設計業務担当

■ 論文要旨 ■

(株) 山下設計では、技術計算に使用していたメインフレームを 1998 年から UNIX サーバにダウンサイジングした。これに伴い、技術計算プログラムを UNIX の環境下で稼働できるように改良した。今回、システムのさらなるスリム化を目標とし、ワーキンググループを結成して比較検討を行った結果、IA サーバを利用した Windows 及び Linux 混在環境への移行を行うこととした。

本論文では、OS の選定理由・システム構築上の工夫点・導入効果について実例をもとに述べる。

■ 論文目次 ■

1. はじめに	《 4》
1. 1 当社概要	
1. 2 構造設計部の業務内容	
2. 現状分析	《 4》
2. 1 現行システムの説明	
2. 2 問題点	
3. 次期システム構成の検討	《 7》
3. 1 システム要件	
3. 2 比較結果	
4. 新システム概要	《 9》
4. 1 構成図	
4. 2 工夫点	
4. 3 移行時の問題点	
5. 導入効果	《 12》
6. 今後の課題	《 14》
7. おわりに	《 15》

■ 図表一覧 ■

図1 現行システム構成図	《 5》
図2 実行画面イメージ (メニュー)	《 6》
図3 実行画面イメージ (ASIST)	《 6》
図4 新システム構成図	《 9》
図5 実行画面イメージ (ASIST)	《 13》
図6 実行画面イメージ (結果表示)	《 13》
表1 テスト環境一覧	《 7》
表2 次期システム比較表	《 8》
表3 処理性能比較表	《 14》

1. はじめに

1. 1 当社概要

株式会社山下設計は、昨年創立 75 周年を迎えた建築設計事務所である。最近の代表作として、大森ベルポート(1996 年竣工)、法政大学ポアソナードタワー(同 2000)、石川県庁舎(同 2002)、六本木ヒルズけやき坂コンプレックス(同 2003)などがある。

現在の所員数は、約 400 名で、そのうち 370 名超が技術者である。

今年、2 月に「リセット 2 1」というプロジェクトを発足させ、営業力の強化、組織力強化を目指し、7 月 20 日から本社を日本橋に移転した。

1. 2 構造設計部の業務内容

山下設計の中で構造設計部は、耐震・免震・制振構造などの様々な構造設計及び耐震診断業務を行っている。現在の部員数は 20 名である。設計では詳細な検討を繰り返して行うため、以前から部門独自のサーバを保有してきた。アプリケーションの内容は技術計算であり、最も多く使用されているのは自社開発の構造一貫設計プログラム ASIST（使用言語：Fortran）である。

従来は、メインフレーム（富士通 VP2100/10）を使用していたが、1998 年から OS に UNIX（Solaris 2.6）を使用したクラスターシステム富士通 AP3000、2001 年からは富士通 PRIMEPOWER に移行し、利用を続けてきた。しかし、運用費用の削減を迫られるとともに、本社移転に向けて、サーバ設置スペースの削減を検討する必要が発生した。

このため、構造設計部内にワーキンググループを結成して今後のサーバのあり方について検討を始めた。

2. 現状分析

2. 1 現行システムの説明

現行システムの諸元は、下記のとおりである。

(1) サーバ

PRIMEPOWER 600 CPU (SPARC64 360MHz) × 4 メモリ 12GB

PRIMEPOWER 200 CPU (SPARC64 300MHz) × 2 メモリ 2GB × 4

計 5 台 12CPU のクラスター構成

OS: Solaris 7

ファイルサーバ: GR720 90GB (RAID5)

バックアップ装置: DLT ライブラリ

(2) クライアント

Windows2000 Professional 23 台

(3) システム構成概要

その構成を、**図 1**に示す。

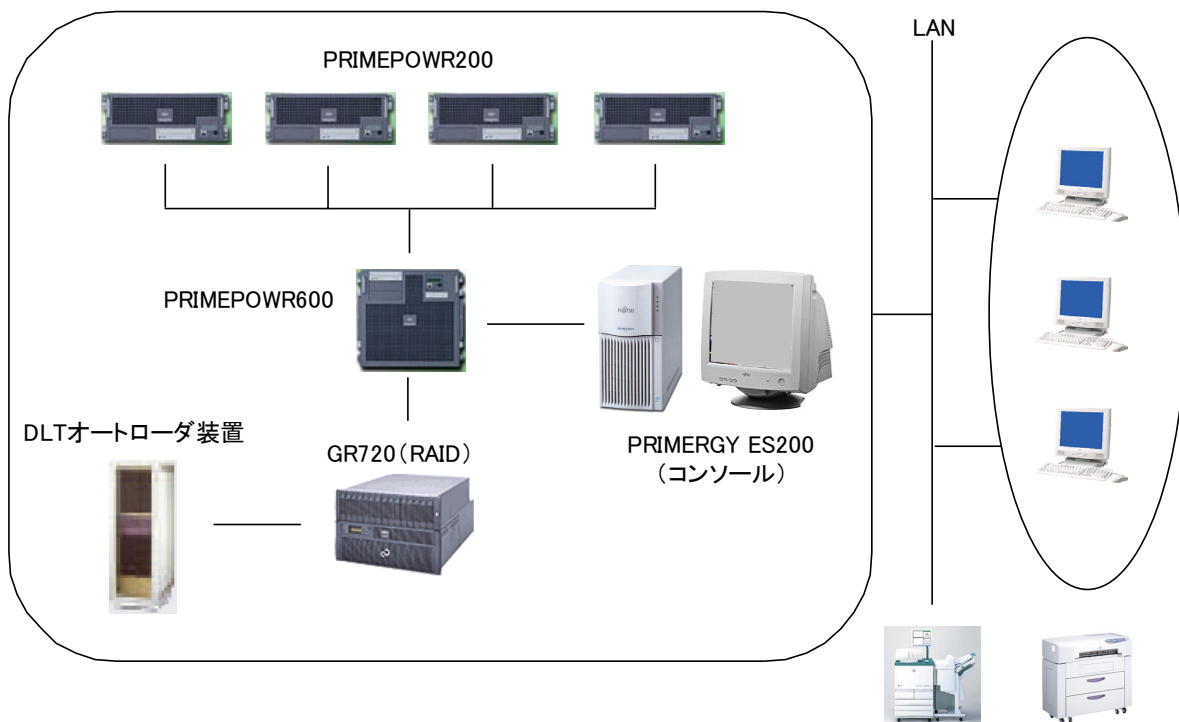


図1 現行システム構成図

- ミドルウェア： LSF (ダイキン COMTEC)
 SafeFILE
 NetWorker
 VisionFS (SCO 社)
- ソフトウェア資産： 自社開発アプリケーション：10 本
 プログラム本数 約 5,000 本 (750,000 ステップ)
 パッケージソフト：6 本
 実行用シェル(Born シェル, C シェル使用)：約 100 本

ユーザーは、各自のパソコン上のエディタからテキストデータを作成する。telnet にてサーバにログインし、シェルスクリプトで作られたメニューから番号を選択して、使用するプログラムやデータを指定する。シェルでは LSF を使用して、各 CPU の排他制御を行い、バッチ形式でジョブを投入している。実行画面イメージを図2、図3に示す。VisionFS を利用することで、各パソコンからはサーバ上のファイルをネットワークドライブ上のファイルとして扱うことができるので、エディタは Windows のエディタ (PowerEDITOR) を使用している。実行結果は、LSF から mail としてはき出されるので、ユーザーが mail コマンドを用いて確認している。

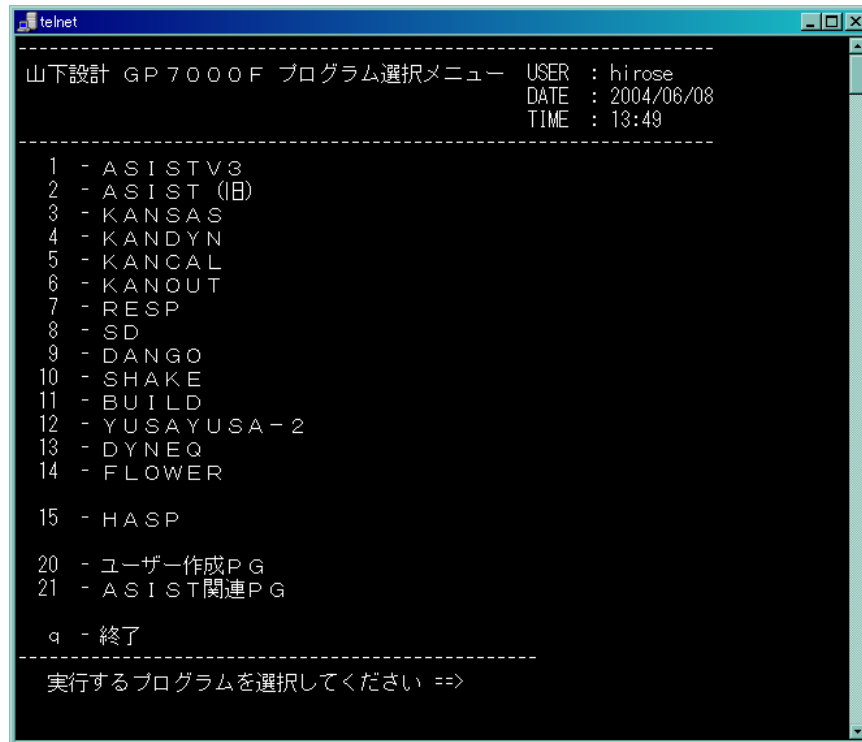


図2 実行画面イメージ (メニュー)

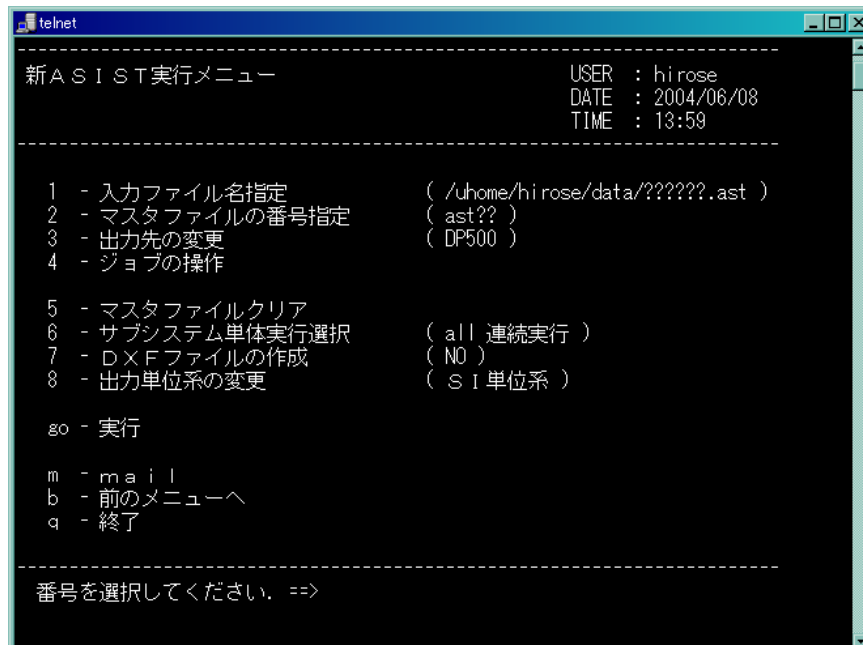


図3 実行画面イメージ (ASIST)

2.2 問題点

第一に、ハードウェアリース料と保守費用の削減が求められている。それと共に、現状ではスペース的にも36Uラック3台とそのほかにバックアップ装置、コンソール装置があり、省スペース化も求められている。

一方で、市販パッケージソフトはここ数年で Windows 化が進んできた。パソコン性能が向上したとはいえ、大きい建物のモデルを解析するため各自のパソコン上で解析ソフトを動かすと CPU 使用率が 100%に上がってしまい他の業務がストップしてしまう。大規模モデルでは、10 分から 60 分以上処理時間を要する。その間は、例えば Excel での計算や、Word での資料作成などを中断せざるをえない現状であった。更に、Windows アプリケーションはバージョンアップが頻繁にあり、Windows Update も含めると管理者に大きな負担がかかっている。

そのほかに、現システムは運用時は安定しているのだが、再起動時にトラブルが多いという問題もあった。これは、クラスタシステムであるため周辺装置も含めた電源投入順序が厳格であるためと推測される。年間を通して、ビルの点検日を除いてはほとんど停止することはない稼働状況であった。

このような問題を解決するべくワーキンググループを発足させ、検討をはじめた。その後、本社移転が決定したので、移転前に新しいシステムに移行を完了する計画とした。

3. 次期システム構成の検討

3. 1 システム要件

前述の問題点を解決するためのシステム要件を整理すると、省スペース、高い費用対効果、技術計算向けに CPU 性能は許される限り最速のものが必要ということになった。ユーザー利用環境は、できるだけ現状から大きく異なることは避けたい。特に市販パッケージソフトが Windows 版であることは動かせないので、前述の CPU 占有の問題を解消するために、Windows のターミナル・サービス機能の利用も検討に入れることとした。

次期システムの OS としては、下記の 3 種類の選択肢が考えられた。

- ・ Solaris
- ・ Windows (ターミナル・サービス)
- ・ Linux

これらの選択肢について導入前の比較検討を行った。そのテスト環境を表 1 に示す。

表 1 テスト環境一覧

ハードウェア : FMV-7000FL Pentium4 1.5GHz 512MB RAM			
OS	日本語 Solaris 7	Red Hat Linux 9 Publisher's Edition (カーネル 2.4) 日経 Linux2003.7 号付録	Windows2000 Professional SP4 (Windows2000 Server と ターミナル・サービス)
コンパイラ	FUJITSU Fortran 90 Compiler V2.0.3 (V4, V5)	FUJITSU Fortran&C Package for Linux V4 評価版	FUJITSU Fortran&C Package for Windows V3.0
実行環境	csh	bash	バッチ プログラム VisualBasic
	telnet	telnet	コマンド プロンプト
開発環境	FUJITSU PC-X V20L31	FUJITSU PC-X V20L32 評価 版	

テスト環境には、特に費用をかけることなく評価版（使用期限付きであるが、機能は製品版と同等）にて本番環境と同等のテストを行うことができた。テストでは特に、現状で最も使用頻度の高い ASIST の動作状況を重点的にテストした。過去に ASIST については Solaris 版でも Fortran コンパイラのバージョンの違いによって動作に問題が確認されたという状況があった。これは、ソースコードに古いルーチンを使用している箇所が多くあるために、例えば、コモン文における配列のサイズにおいて文法を厳格に守っていないような箇所がコンパイラのバージョンにより動作が異なってくるためと推測される。これまで、コンパイラのバージョンを上げるとサブルーチン間で、引数が正しく渡らないためにプログラムが異常終了したり、結果が大きく異なることが発生していた。計算は、技術計算であるので小数部分までの高い整合性が要求されていた。

3. 2 比較結果

各項目別の比較テスト結果を表 2 に示す。なお、現行の Solaris 環境を基準として比較している。

表 2 次期システム比較表

		Solaris	Linux	Windows
Fortran 資産 の移行	ソースコード	V2 では問題なし V4 以降では、問題あり	互換性が高い ソースレベルでの修正は不要	一部修正が必要 プログラムサイズの制限を超える
	make ファイル		1 箇所のみ修正	つくり直し
	デバッガ		操作性に互換性なし	ほとんど同じ
シェルスクリプト (バッチコマンド)			互換性が高い echo 文を修正	同等の動きはできない
プリンタ出力（文字とオーバーレイ出力のみテスト、図形出力は未テスト）			問題なし	コード体系が異なるだけで他には問題なし
操作性	telnet コマンドプロンプト		同じ	大体似ているが一部異なる
	X Window		ほぼ問題なし 日本語入力に問題あり	—
	PC-X		セキュリティ設定の変更が必要	—
管理			同等 セキュリティ設定がより詳細にできる	一部問題あり
ファイル共有		VisionFS	Samba サーバ 同等	問題なし
パフォーマンス			問題なし	問題なし
LSF			同等	一部異なる

最も問題となったのは、ASIST を Windows に移行した場合の実行方法である。まず、コマンドプロンプトから実行するようにバッチコマンドで作成してみたが、シェルスクリプトのように柔軟性がなく、かなり無理があることがわかった。次に、VisualBASIC で実行用ウィンドウを作成してみたが、ASIST 実行後のリターンコードを取得することができないことがわかり、実行後のエラー処理に問題が発生した。このため、ASIST を Windows に移行するのは断念し、Linux を選択することにした。

Linux はバンドルされている Samba サーバなどのパッケージが豊富で、結果的に費用削減につながるということがわかった。特に、Solaris から移行する場合の互換性は予想以上に高かった。テスト時の問題点として、Samba サーバの構築、日本語入力、セキュリティ設定について安定性の点で不安があったが、これは、企業向け製品版を導入し、販売元またはディストリビュータのサポートを受けることで解決できることが期待できた。

Windows 環境では、MetaFrame^{注1)} については、簡単にテスト環境を構築することはできないためターミナル・サービスの機能テストのみに留まったが、利用イメージはつかむことができた。また、ターミナル・サービスと MetaFrame の機能の差については、雑誌や HP から得ることができ、問題となりやすいポイントについても把握することができた。

4. 新システム概要

4.1 構成図

新システムの構成図を図4に示す。

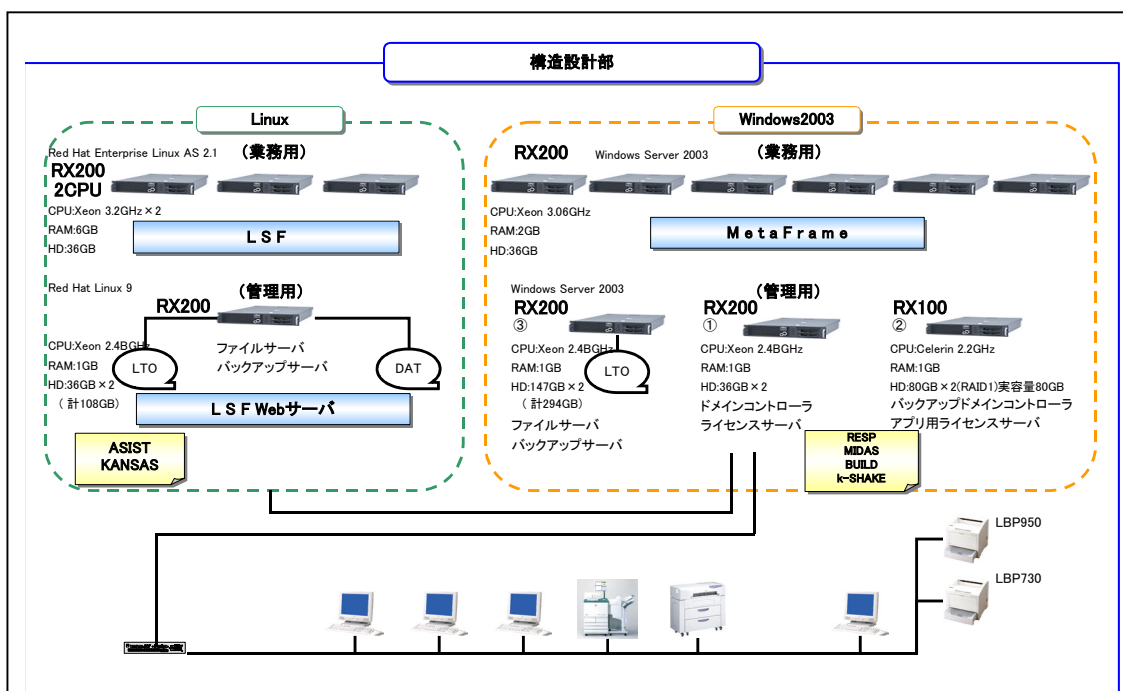


図4 新システム構成図

注1) ターミナル・サービスはWindowsサーバOSに標準で実装されている機能で、クライアントPCから、サーバ上のアプリケーションや管理ツールなどを実行することができます。MetaFrameはCitrix社が提供している製品で、ロードバランシング機能や管理機能・セキュリティ機能が付加される。

搭載するアプリケーション

Linux サーバ

- ・ ASIST (構造一貫設計プログラム)
- ・ KANSAS (有限要素法プログラム) 注2)
- ・ その他ユーザー作成プログラム

Windows サーバ

- ・ RESP シリーズ (振動解析プログラム) 注3)
- ・ MIDAS/Gen (有限要素法プログラム) 注4)
- ・ SEIN La CREA (構造一貫設計プログラム) 注5)

4. 2 工夫点

前述の比較結果から ASIST は Windows 環境に移行することは互換性の点から問題が大きいと判断し Linux を選択した。ASIST はメモリ容量が処理速度に大きく影響するため、このモデルに搭載可能な最大の 6GB のメモリ構成とし、これを使用するため、OS は Red Hat Enterprise Linux AS を選択した。

市販パッケージソフトは Windows サーバ上で MetaFrame を使用して負荷分散を行い、CPU 占有の問題の解決を図ることとした。このため、新サーバの OS は混在環境となることになった。MetaFrame を使用することで高価な高性能サーバを数多く導入することなく資産を有効活用することができる。管理者側からは、アプリケーションソフトの管理が従来より容易に行えるようになる。

ハードウェアには価格性能比が高く、かつ、1U サイズ (厚さ 44.45mm) である IA サーバを選択した。これは、将来的に ASIST と市販パッケージソフトの使用頻度が変わってきたときに、OS を再インストールすることで容易に転用できるという柔軟性も考慮した。出力機器は、高速プリンタ (オーバーレイ使用)、静電プロッタとも従来から引き続き使用できる。

さらに、Linux 側には、Web Launcher を導入することとした。これは、LSF にジョブを投入する際、従来は、シェルスクリプトで作成したメニューから行っていたものを、Web 上から行えるようにしたもので、ユーザーはマウス操作でジョブ投入から実行状況の表示、結果の確認まで行えるようになる。

なお、NAS サーバの導入も検討したのだが、ハード/ソフト費用が高くシステム全体に占める割合が大きくなってしまふことから断念した。Samba サーバで機能的に代用することとし、費用を圧縮した。

注2) KANSAS : 鹿島建設 (株)

注3) RESP シリーズ : (株) 構造計画研究所

注4) MIDAS/Gen : (株) 構造計画研究所, MIDAS 社

注5) SEIN La CREA : (株) NTT データ

4. 3 移行時の問題点

(1) Fortran 互換性

テスト時には見つからなかったのだが、16 進定数の扱いが Solaris 版と Linux 版で異なることが判明した。日本語文字列を指定する際にこの 16 進定数を使用している箇所があった。これは、該当箇所のソースコードを修正することで対応した。それ以外にも、未定義変数の扱い方に差があるなどいくつかの修正箇所が見つかった。

(2) 現行 DLT 資産の移行

現行システムのデータバックアップは毎週 1 回 NetWorker から DLT テープに行っていた。以前の AP3000 からの移行時には、同じ Solaris 同士であったので、テープは変換せずそのまま引き継いでいた。今回は、更に大容量高スピードの LTO 装置を導入することになったので、変換が必要となった。対象となるテープは過去 6 年分の計 15 本（1 本 80GB）が保管されていた。変換作業をはじめると、このうち 3 本のテープが媒体エラーで読み取り不可となってしまった。これは予想以上に信頼性が低かった。また、ディスクにリストアする手順は、コマンドベースで使い勝手が悪く、その都度 1 本のテープを全読みするため非常に時間がかかった。（1 回につき約 3 時間）このため、5 年分 9 セット分を残すに留まった。

(3) シェルスクリプトの仕様の違い

テスト時には見つけられなかった細かい仕様の違いがあった。（例えば、* や く の解釈の違い）これは、実際に動かしながら 1 箇所ずつロジックを見直し、手修正した。最終的には、動作には問題ない変換が完了した。

(4) Linux と Windows におけるユーザー管理の整合性

一人のユーザーを Linux ユーザー（NIS:Network Information Service）、Samba ユーザー、Windows ActiveDirectory ユーザー（MetaFrame ユーザー）としてそれぞれに登録が必要であり、その手順を確立するまで、試行錯誤があった。

(5) Linux サポートについて

インターネット上で、Linux 関連情報（Red Hat ディストリビューション以外も含む）は豊富に得ることができるのだが、Solaris からの移行という観点での情報は意外に見つからなかった。特にシェルの仕様の違いなど今回は手探りの状態であったので、今後他社からも需要は高いと思われる。富士通サポートデスクにてサポートを受ける際も、OS 別の切り口になっており移行としてのサポートの必要性を強く感じた。また、サポートセンターに問い合わせる際、質問をしてから初めてそれはサポート対象外という回答を返され移行作業に支障を来すということが何度かあり、Linux サポート範囲に何が含まれていないのかがあいまいであると感じた。

(6) MetaFrame とファイルサーバの関係、アプリケーションとの関係

テスト時には、前述のようにターミナル・サービス環境までしか調べることができなかった。このため、MetaFrame にアプリケーションを登録する場合の個別の最適な設定方法は、アプリケーションごとに動作状況を確認しながら、表示方法や環境設定ファイルの保存場所などを決めていった。その他、プリンタ登録・印刷出力関連もテスト時にチェックできなかったのが不安があったが、特に問題は発生しなかった。

(7) アプリケーションの問題

Windows サーバに載せる予定であったアプリケーションの一つ（MIDAS/Gen）が、本番環

境で動かすと、期待した性能が出ない現象が発生した。調べたところ、解析処理中に作成しているワークファイルをデータファイルがあるディレクトリ上に作成しているため、今回の構成ではネットワーク間で多くの I/O が発生する結果となっていたためと判明した。現状のバージョンではワークファイルを作成する場所を変更することができないため、販売元に変更を依頼した。その問題が解決するまでは、このアプリケーションについては従来通りローカルのパソコン上での運用とすることにした。

5. 導入効果

(1) 費用・スペース削減

費用は、ハードウェア及びソフトウェアの月額費用（4年リース）ベースで1/3に削減できた。同サポート費用は1/2となり、これらを合わせて当初の目標を達成することができた。

スペース的には、今まで必要だったスペースは 36U ラック 3 台、DLT ライブラリ装置、コンソール用 PRIMERGY であったものが、36U ラック 1 台、コンソール用ディスプレイ、キーボードとほぼ3分の1に減らすことができ、これもラック 1 台に収めるという目標を達成することができた。

(2) 業務効率（ユーザーの使い易さ）の向上

Windows アプリケーションについては、MetaFrame を介していることはほとんど意識せずに行うことができる。注意が必要なのは、プログラムのハングアップ時にはログオフ処理によってセッションを切断する必要があることである。ほかにはローカルで実行する場合と、使用方法の違いはなくストレスなく解析処理を行うことができるようになった。

Linux 側は、ASIST については Web Launcher からの実行となったので従来より使い勝手よく結果確認を行えるようになった。利用イメージを図 5、図 6 に示す。



図5 実行画面イメージ (ASIST)

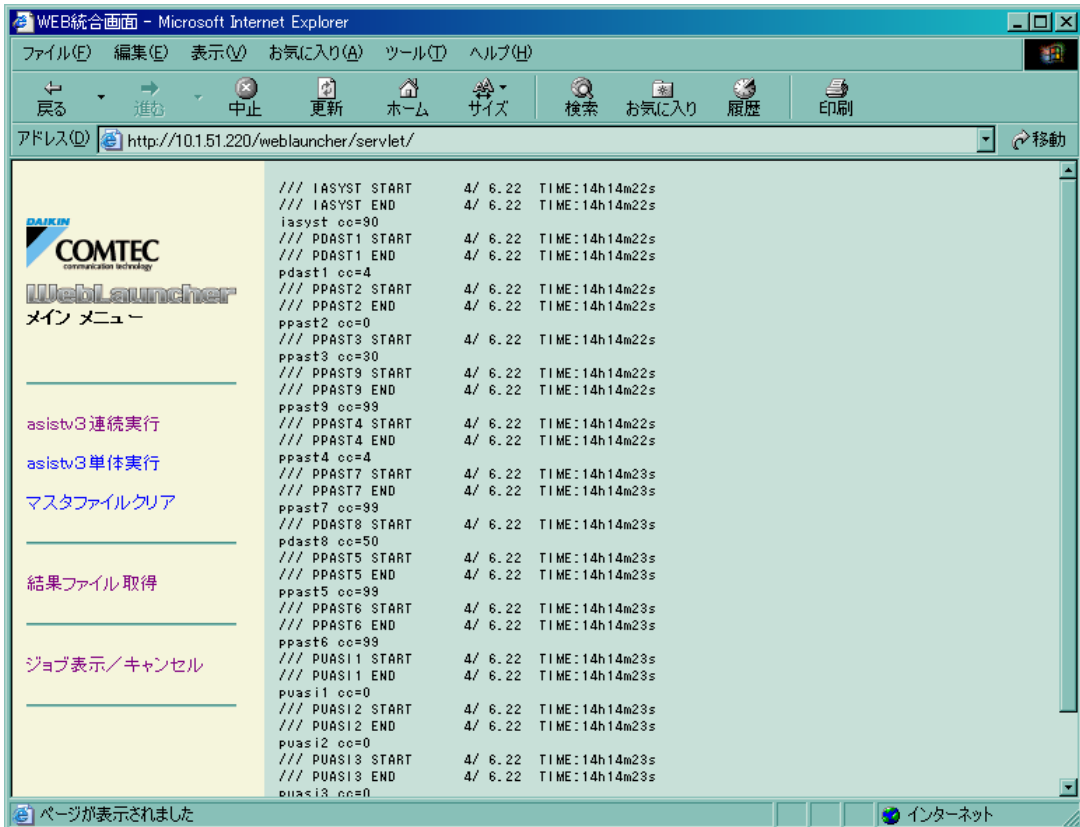


図6 実行画面イメージ (結果表示)

(3) 処理性能の向上

処理性能は、機器購入前にカタログベースで比較しても、なかなか実態はつかめないことが多い。動かすアプリケーションの性質によって、結果が大きく変わることもある。今回、実際に同じアプリケーションを実行して比較した結果は満足できるものであった。これは、CPU 性能だけでなく、メモリを大きくしたこととサーバ間の接続を全てギガビットにしたことが、効果的であったと推測される。参考までに比較結果を**表3**に示す。

表3 処理性能比較表

	Solaris	Windows(ローカル)	Windows(サーバ)	Linux
ASIST	4137.43	742.67	—	419.35
KANSAS	85.94	—	—	8.87
MIDAS(解析処理)	—	6.0	3.6	—
RESP	59.0	—	23.0	—

単位：秒

太枠内は、新サーバでの実行環境

注) 各 OS と CPU

Solaris : Solaris7 SPARC64 360MHz 12GB RAM

Windows (ローカル) : Windows2000 Pentium4 1.5GHz 512GB RAM

Windows (サーバ) : Windows2003 Server Xeon 3.06GHz 2GB RAM

Linux : Red Hat Enterprise Linux AS 2.1

Xeon 3.2GHz 6GB RAM

(4) Windows アプリケーションの一括管理

Windows アプリケーションにおける頻繁なバージョンアップの度に各ユーザーのパソコンを更新しなければならなかったが、今後は6台にインストールするだけでよいので、手間は1/4となった。

6. 今後の課題

(1) メモリ不足

今までは、年間を通してほぼ連続運転をしていたが、移行後 Windows サーバは週に1回定期的に自動再起動を行うように設定した。これは、予期せぬメモリ不足の発生を防ぐためである。

一方、Linux 管理サーバは1GBのメモリ使用率がWeb Launcherの使用率によっては不足してくることも予想されるような状況なので、使用率を見ながら拡張を検討することとした。これについては、IAサーバであることから、費用も数万円程度に収まり、入手後、簡単にセットできるので、必要になればすぐ対応することができるものとする。

(2) バックアップ処理運用体制

今回は、費用の制限からディスクをRAID構成にすることができなかった。しかし、データの性格上テキストデータから計算結果を復元することは容易にできるので、大きく

問題になることはないと思われる。バックアップソフトウェアと装置については費用を抑えたため、ある程度手動で行う部分が発生するが、運用でカバーしていきたい。

(3) 管理が二重化

Windows と Linux で、それぞれの管理が必要であるが Linux は、Solaris のオペレーションと比べて、違和感なく使用できた。今後は、Windows と Linux のユーザー管理・ファイル管理について運用マニュアルを作成し、うまく整合性をとっていきたい。

(4) Linux 利用環境の向上

今回、費用と時間的な制限から、ASIST 以外のアプリケーションについて telnet からの利用環境が残ってしまったのだが、将来的には Linux 上のアプリケーションはすべて Web Launcher から利用できるよう順次切り替えていきたいと考えている。現状では、Web Launcher にトラブルが発生した時のバックアップ用に従来のシェルスクリプトでの実行環境も残している。基本的には、Web Launcher からジョブ実行用シェルスクリプトに制御を渡す仕組みであるので、併用や切り替えには大きな問題はない。

7. おわりに

今回のシステム構築は、本社移転というタイムリミットがあり最後の半年間は、スケジュール的にも非常に厳しいものであった。しかし、1年以上前からワーキンググループとして取り組んできた構造設計部のサーバ刷新に向けての問題点の洗い出し、さまざまな比較検討作業を通して、最善の選択をすることができたものと信じている。比較テストを実施するに当たって、各ソフトの評価版を無償で入手することができて本番環境に近い状態で動かしてみることができた。これにより、実際に使ってみて、要求している機能が満たされるのかチェックしてから正式に導入できるので、ユーザーサイドから見ればありがたいことであるし、ソフトウェアの選定をベンダーまかせにせず、チェックフェーズを重視するという姿勢は成功の鍵ではないかと感じた。

また、本論文作成を通して、今回の一連の成果をまとめることができたことは、今後の業務に生かしていく上でも、貴重な経験となった。同時に、これから Linux の採用を検討しようとする企業、組織の方々の参考になれば幸いである。

最後に、ご協力いただいた富士通株式会社殿、株式会社富士通システムソリューションズ殿をはじめ関係者の方々に深く御礼を申し上げるものである。