
Webシステム構築について

—公文書管理システム再構築を通して—

(株) ティー・アイ・シー

■ 執筆者Profile ■



船越 悟

1994年 (株) ティー・アイ・シー入社
システム開発業務担当
(予算編成・決算書・決算統計等)
1995年 文書管理システム(C/S・16ビット版)開発に参加
1996年 文書管理システム(C/S・32ビット版)
パッケージアプリケーションとして開発開始
1999年 公文書管理システム(Web版)開発開始
2001年 現在 オープンソリューションシステム部
パッケージグループ 所属



塚本 達二

1999年 (株) ティー・アイ・シー入社
システム開発業務担当
2000年 パッケージ開発業務担当
2001年 現在 オープンソリューションシステム部
パッケージグループ 所属

■ 論文要旨 ■

現在弊社で開発・販売中のクライアント/サーバ(以下 C/S)版公文書管理システムのユーザから、イントラネット環境に対応した文書管理システムの要望があり、開発を行った。当初は大規模にも対応できるシステムを目指し、Javaをベースとして富士通製アプリケーションサーバ「INTERSTAGE」とオラクル製データベース「ORACLE8i」をプラットフォームにシステムを構築した。

その後、比較的小規模なユーザのニーズに対応すべく、コストダウンを図り、フリーWebサーバ「Apache」等を中心にした弊社独自のアプリケーションサーバ形態の構築を行った。現在は今後の電子決裁・電子文書等の電子行政をにらみ、プログラムの部品化(Java Beans等)を進め、UNIX・Linux対応版の開発に着手している。スケラビリティに優れた「統合型」公文書管理システムを目指している。

■ 論文目次 ■

1. はじめに	《 3》
1. 1 当社概要	
1. 2 公文書管理システムの特徴	
2. 既存C/S版の問題点	《 3》
3. システム開発	《 4》
3. 1 システム設計	
4. 開発の現状と課題	《 6》
4. 1 開発の現状	
4. 2 開発の課題	
5. 今後の展望	《 8》
6. おわりに	《 8》

■ 図表一覧 ■

図 1 論理設計図	《 4》
図 2 システム技術概念図	《 6》
図 3 新システム画面（収発分文書登録業務）	《 6》
表 1 負荷テスト（物理2層その1）	《 9》
表 2 負荷テスト（物理2層その2）	《 9》
表 3 負荷テスト（物理3層その1）	《 10》
表 4 負荷テスト（物理3層その2）	《 10》

1. はじめに

1. 1 当社概要

当社は昭和49年設立。受託計算センター業務を中心として成長を続けてきた。現在では官公庁向けシステム、パッケージ開発、ネットワーク関連システム構築、データベース設計、ハウジングサービスなど、PC から汎用機までに対応する総合的な IT ソリューション・ビジネスを展開している。

1. 2 公文書管理システムの特徴

各自治体がそれぞれ管理している文書は、年間で数万から数十万以上発生している。さらにそれらの文書を整理し、保管・保存・廃棄といった一連の作業を人の手で行うには、相当のコストと時間を費やせねばならない。そこで、文書管理の改善を行い、管理方法として代表的な「ファイリングシステム」注1)¹を導入し、文書の整理を行い、管理方法を確立し、その電算化として紙文書の発生から所在管理を LAN 上で効率的に管理するシステムが求められてきた。

そこで当社はまず、比較的開発・導入の容易な Microsoft Visual Basic (以下 VB) で開発したクライアントアプリケーションと、Microsoft SQL Server (以下 SQL Server) をデータベースとしたクライアント・サーバ (以下 C/S) 型の公文書管理システムを構築した。これは論理的・物理的 2 階層 (下記参照) の形態である。

その後、1999 年末に既存の C/S 版ユーザから、Web 対応の公文書管理システムを受注したことから、Web 対応版の開発に着手することになった。

<階層について>

文中に「階層」という言葉を頻繁に使用している。これはシステムをどの段階で何を処理するかという処理単位での意味で、「論理 X 階層」と表記する。また、物理的な位置を「物理階層」と表記する。どちらもクライアント側からサーバ側へと順番に第 1, 2 と階層が進む。本文では、プレゼンテーション層 (第 1 階層)、アクセス層 (第 2 階層)、アプリケーション層 (第 3 階層)、データベース層 (第 4 階層) と記述していることにも注意されたい。

2. 既存 C/S 型の問題点

VB はその名の通り非常に視覚的に開発を行え、経験の浅いプログラマ初心者から上級者まで世界的に普及している開発言語である。SQL Server も同様に、Windows 環境での親和性の高さでシェアを伸ばしているデータベースである。

しかし、現在ではその優位性も薄れ、以下のような問題が挙がっている。

1) Windows (OS) 依存

VB は DLL や OXC などの Windows コンポーネントを利用するため、最近のオペレーション・システム (以下 OS) の頻繁なバージョンアップに対してメンテナンス性の面において脆弱である。

2) クライアント単位での環境構築の必要性

大規模なシステムを構築する際に、各クライアントへのセットアップ及び動作設定・確認が煩雑であり、コスト高の傾向がある。

3) インターネットの普及

昨今のインターネットの爆発的普及は、自治体にも波及し、多くの自治体でインターネットとシームレスな連携が可能なシステムを考えるようになってきている。現状では対応が困難。

以上の点に加え、従来型の2階層のC/Sでは、データベースはサーバ上にあり、アプリケーションのロジックとユーザーインターフェースはクライアント側に渾然一体となって同居し、プレゼンテーション層とアプリケーション層がクライアント側で実行される。この構成だと、データ管理層とアプリケーション層がサーバとクライアントに分かれるため、ネットワーク上に大量のトラフィックが発生してしまう。さらに、負荷の大きな処理をクライアントのCPUで処理するのだから、これは根本的に無理がある。この問題を解決するために考えたのがn階層のC/Sである。

3. システム開発

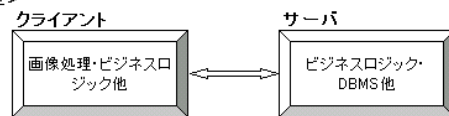
3.1 システム設計

3.1.1 論理設計

新システム開発は、既存のシステムをいかに有効に活用して、再構築できるかが焦点となる。従来のC/S型はクライアントサイドに重点を置いているため、まずはクライアントに必要なリソース(資源)を少なくさせることに着目した。論理設計段階では、機能を分割して多層化し、負荷分散も容易な構成を考慮した。(図1を参照。)

システム比較

<従来型>



<n階層>

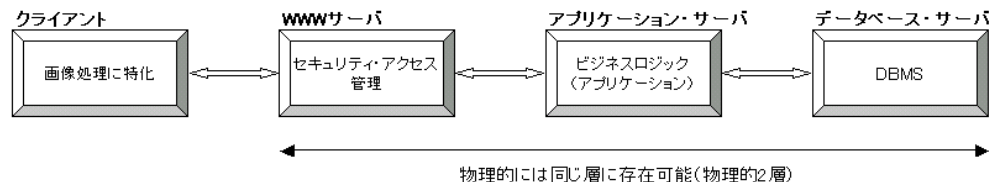


図1 論理設計図

結果的に、従来型では論理2階層・物理2階層の構成を、新システムでは論理4階層・物理2～4階層と細分化した。よく3階層という分類を耳にするが、当社ではクライアントからの要求を管理する部分を階層化している。セキュリティや負荷対策として、WWWサーバの担当する階層は分散配置すべきであり、ビジネスロジックはクライアントあるいは侵入者などから隠蔽される方が良いと考えたからである。

3. 1. 2 機能・技術設計

開発技術の選択は、非常に重要である。既存システムからシフトするならば、本来ならHTML+ActiveX+SQLServer+ α となるのだが、これでは汎用的ではなく、Windows依存の環境から脱却できなくなる。そこで、プレゼンテーション層の技術として、HTML、Javaスクリプトで画面を作成し、サーバサイドはJavaServlet+Javaパッケージをアプリケーション層に、JDBCⁱⁱ+Oracle8iをデータベース層に利用することで、LinuxなどのOSへ移植できる汎用性を確保した。

文書管理においてクライアント層に必要な機能は、大きく2つに絞られる。

- 1) 文書の登録・編集，検索
- 2) 起案書などの帳票を表示・印刷

これらを既存システムでは1)をVBが、2)をSeagate社製のCrystal Reportsを使用していたが、それぞれInternet Explorer(以下IE)とAdobe社製Acrobat Reader(以下AR)を利用することにした。これらはほとんどのクライアントの標準環境で存在し、かつ無償である点が最大のメリットである。Netscape Navigator(以下NN)も候補ではあったが、起動時間、Javaスクリプトの実行速度がIEより劣る点が致命的であった為、採用を見送った。ARはデファクト・スタンダードのPDF形式で表示可能な唯一のプラグインビューワーである。

また、サーバサイドに必要な機能は、クライアントの要件に加え、

- 3) 各クライアントのアクセス監視
- 4) 要求の高速な処理
- 5) スケーラビリティ(拡張性)
- 6) 開発環境の充実

が必須である。アクセス層とアプリケーション層に富士通製のアプリケーションサーバ「INTERSTAGE V3」を採用した。INTERSTAGEは、EJBⁱⁱⁱやCORBA^{iv}をサポートしており、専用の統合開発環境「Apworks」による容易な開発が可能である。データベース層にはOracle8iを、帳票出力エンジンとして翼システム製「SuperVisualFormade」をそれぞれ採用した。これらはシェアの高さと技術情報の入手の容易性を評価した。

開発言語については当然Javaを利用するが、JavaのAPIであるJDK(Java Development Kit)も当初1.1系が安定版であったことから、JDK1.1.8を採用した。

当面の需要としてはWindowsベースがほとんどで、しかも開発期間が初期段階で3～5ヶ月と短かったことから、サーバOSにはWindowsNTServer4.0を採用した。(図2参照)

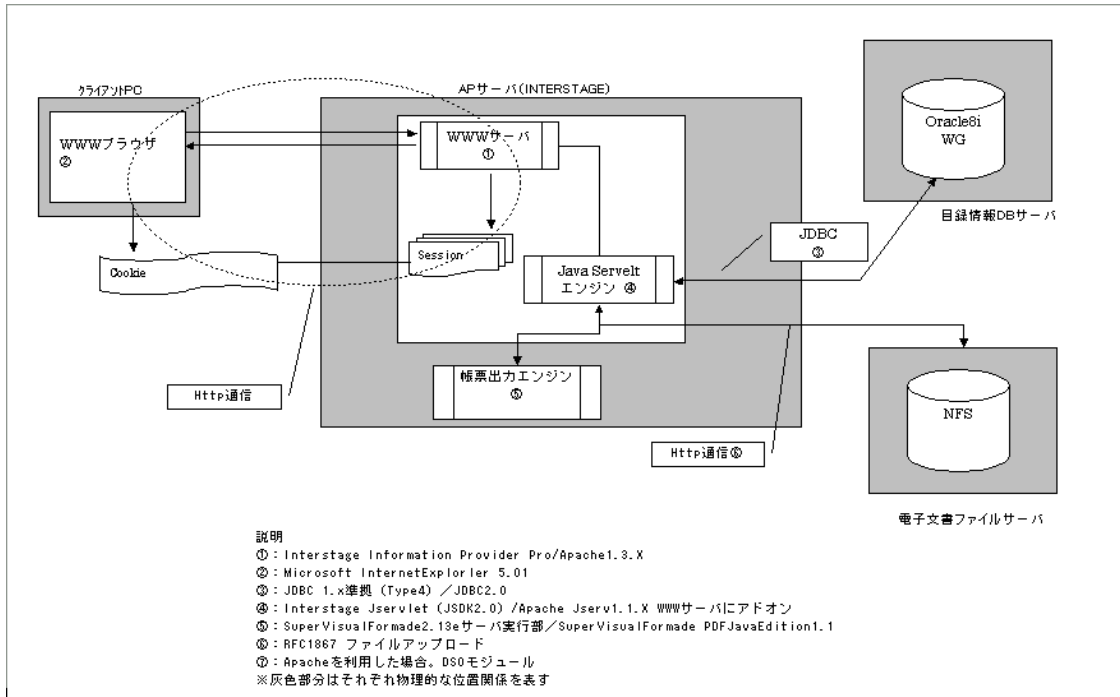


図2 システム技術概念図

4. 開発の現状と課題

4.1 開発の現状

4.1.1 工夫点

まず画面について、800×600ドット (SVGA) サイズで、極力スクロールが生じないように設計した。これは旧資産の活用と、文書を登録・編集する収発文書登録画面などでの操作性を、既存システムの画面に近づけるためにも必要であった。特に画面を重ねる「タブ」機能を実現させるためには、VBでは専用の部品により容易だったが、新システムでは、スタイルシートやHTMLを駆使しての実現となった。(図3参照)

収発年度 平成13年度 所属 9999999 TIC

収発番号 TIC発 表示 発番

文書件名

入力区分 収发文書 発議文書 供覧文書 資料

収受日 令連件番号 第 番

処理期限 起案日 決裁日

発信日 施行日 完結日

担当者・起案者 / 相手方・発信先 / 合議・行為 / 説明 / 保存年限・公開区分

担当者 起案者

担当者TEL 担当者入力 起案者TEL 起案者入力

平成13年度 ガイド検索

共通固有区分

共通文書

各課固有文書

バインダー・簿冊

第1ガイド

第2ガイド

第3ガイド

フォルダ

新規登録 更新 削除 完結 未完参照 印刷 画面クリア

図3 新システム画面 (収发文書登録業務)

さらに、文書管理システムにログインしているユーザの操作時間を WWW サーバ側で設定して、セッションの有効期限と共に管理するようにした。同時に接続するユーザ数も可変として、これにより、Oracle 同時接続ライセンスのコスト削減と、無駄なアクセスでの負荷を減少させることが可能になった。

Java Servlet を利用する際に最も問題となるのが、日本語表示の文字化けである。Java は基本的に内部変数は Unicode に変換されるために、クライアントのブラウザの文字コードを意識した変換を行わないと、正しく文字がエンコードされない。当社でも例外に漏れず、大変な労力を費やす結果となった。特に 2 バイト文字の記号「～」などは、Windows 側の MS932 系統の内部マッピングと、Java 側の Unicode の JIS 系統の内部マッピングでは変換テーブルが異なるために、それらの文字は独自のクラスライブラリを開発して変換する必要が生じた。

統合開発環境「Apworks」は、HTML の画面作成から JavaServlet までを容易に作成できるため、画面デザイン→画面作成→JavaServlet 作成→データベースアクセスクラス作成→デバッグと一連の開発を効率よく行えた。ポーランド製の Java 統合開発環境「JBuilder」と比較しても、動作が軽く、ヘルプが見やすい。ソースの自動生成機能により、Java での開発経験の乏しい当社のスタッフも、十分な戦力となることができた。

4. 1. 2 フリーウェアの実用性

IT 関連業界の宿命として、度重なる技術革新のサイクルは、最近急速になる一方である。新システムがアプリケーションサーバ依存の形態ではなく、パッケージソフトとしての幅を持たせ、より小規模なユーザに対応する必要が生じてきた。アプリケーションサーバの利点である拡張性は反対に、コスト面でのデメリットがある。オープンソースが主流になっている現在、WWW サーバで圧倒的なシェアを誇るフリーの「Apache」を採用して、Java Servlet コンテナ「Apache Jserv」を組み合わせ、独自のアプリケーションサーバ形態を開発した。（**図2**参照。INTERSTAGE 部分の枠が無い構成）基本は INTERSTAGE V2 の WWW サービスと Servlet サービスで、それらは Apache と Jserv が原型となっている点をヒントにした。当然ロードバランサーや TP モニタなどの高度な負荷分散機能は有していないが、Apache には DS0形式のモジュールをアドオン（追加）し、機能強化を行うことが可能になっている。ほとんどのプログラムモジュールは Apache+Jserv 環境化でも動作した。一部 java.io パッケージのファイル読み出し関連のパスの指定が、絶対パスでないという現象が見られたが、重大な問題ではないと判断している。実際にユーザの運用でも Windows サーバ1台で同時50台程度のアクセスにも耐えるという結果から、人口が約10万人以下の自治体では Apache+Jserv 形態のシステムで、それ以上の場合に INTERSTAGE を利用するシステムといった構成で提案を行えるようになった。

4. 2 開発の課題

4. 2. 1 システム負荷

第1番目のユーザへ INTERSTAGE 利用の新システムを納品する時に、社内において負荷テストを行った。（詳細は付録表1～4「負荷テスト」を参照）負荷テストの結果、物理的に2階層よりは3階層の方がより負荷に強く、また同じ3階層でも、特に論理4階層のデータベース層のサーバ性能が良い方が、負荷に強いことが判明した。もちろん負荷テスト

を行った処理部分にもよるが、大量の帳票を出力した場合はアプリケーション層に大きな負荷がかかるなど、運用状態により異なる負荷状態が存在することは考慮すべきである。さらに、2000年12月に行った負荷テストでは、ウイルスチェックアプリケーションをサーバで実行すると、最大で25%程度の性能低下が認められている。

負荷分散については、各サーバのハードウェア性能をアップさせる（スケールアップ）か、サーバ自体の追加を行う（スケールアウト）等の対策が広く知られている。

それだけでなく、システム内部の最適化（コーディングの見直し、データベースの再構築等）と並行してチューニングを行う必要がある。

4. 2. 2 その他の問題

新システムではプレゼンテーション層をHTML+Java スクリプトに絞ったために、表現力及び操作性がIE依存になってしまっている。これが大きな問題になりつつある。IEは4.01SP2よりサポートを開始したが、バージョンにより仕様が微妙に異なり、その対応には相当の人員を投入しなくてはならず、それによってシステムの版数管理も非常に煩雑なものになってしまっている。

アプリケーション層の構造でも、現在はビジネスロジックとプレゼンテーションロジックが混在しており、今後はJavaServerPages（以下JSP）の導入も視野に入れ、ロジックの分散化を進めていくべきだろう。

5. 今後の展望

新システムには大別して3つの機能を包含する予定である。具体的には文書管理（目録情報）と、電子文書管理（添付ファイル方式）、そして電子決裁（ワークフロー）を指す。グループウェアとの連携を考え、人事情報の共有化に備えるために、データ共有化のインタフェース開発も予定している。それに加えて、総合行政ネットワーク（LG-WAN）で標準とされるXML形式の文書を取り扱えるように、XMLパーサの実装をApacheに組み込む実験を開始している。当然マルチ・プラットフォーム化をにらんでいるが、現在はRedHatLinux系のディストリビューションで動作確認中である。UNIXへの移植は今後の動向を見定めてからになるであろう。

6. おわりに

今回の文書管理システムの開発に関わって、n階層を正確に把握することにより、ロジックの分散化は容易であり、開発工数の短縮になるというメリットがあるということを実証できた。さらに、汎用的な技術を利用している限り、ひとつのプラットフォームに依存しない、スケーラビリティに優れたシステムを構築できることの有用性を、再認識できたことが大きな収穫である。お客様はより多くの選択肢を得る必要があり、我々はより多くの選択肢をお客様に提供・提案する必要がある。今後はさらなる効率的かつ汎用的なシステムの構築を目指したい。

付録

表1 負荷テスト（物理2層その1）

Web対応文書管理システムのサーバ負荷テスト報告

実施日：平成12年8月17日～18日

タイプ1（2層：Webサーバ・DBサーバ兼用型）

①サーバ01（DELL）

ハードスベック
 CPU Pentium III 500Mhz
 MEMORY 256MB
 PageFile 初期 384MB 最大 1024MB
 HDD 9GB(C:4GB、D:5GB)
 OS WindowsNT4.0 SP6a
 アプリケーションサーバ
 INTERSTAGEV3.0L1.0(富士通)
 データベース
 Oracle8i WorkgroupServer 8.1.6(オラクル)
 Webサーバ
 Information Provider Pro(富士通)
 Java
 JSDK2.0(Sun)
 JDK1.2.2(Sun)
 帳票出力アプリケーション
 SuperVisualFormade Ver2.13e(霞システム)
 Javaクラスライブラリ Ver1.00a
 使用テストツール [Microsoft WCAT Ver4.35]

クライアント数/台	応答時間/ミリ秒※1	CPU使用率/%※2	発生エラー合計/個※3
50	100998.7/12037.7/0.0	98.814	6
45	218094.0/10403.0/0.0	97.821	4
40	145399.0/12124.0/0.0	98.931	4
35	83855.5/9742.0/0.0	98.334	4
30	97480.5/11273.0/0.0	98.079	2
25	97415.0/7100.0/0.0	98.117	1
20	95027.0/6852.5/30.0	98.104	0
15	72323.5/5229.0/35.0	98.232	0
10	16879.5/2284.5/20.0	95.695	0
5	6830.0/1132.0/20.0	72.148	0

※1 最大/平均/最小 正確に計測できない場合は省略

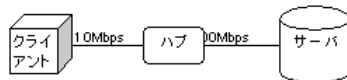
※2 WinNTの「パフォーマンスモニタ」使用 300秒のテスト前後10秒は誤差として計算外とした

※3 テスト使用プログラムでのエラー数の合計値 基本的には平均値(小数点以下四捨五入)

②クライアント01 (Fujitsu FMV-5233NU/W)

ハードスベック
 CPU MMXPentium 233Mhz
 MEMORY 64MB
 OS WindowsNT4.0 SP5

③ネットワーク環境



負荷テスト処理内容

- ①コードメンテナンス更新処理 1件 40%
- ②収発入力業務 表示(検索)処理 1/100000件 30%
- ③収発入力業務 起案書出力(プレビュー)処理 1件 30%
- PDF形式ファイル 出力サイズ:12KB
- 使用スクリプトファイル xxx1.scr, xxx1.dst, xxx1.cfm

表2 負荷テスト（物理2層その2）

実施日：平成12年8月21日～22日

タイプ2（2層：Webサーバ・DBサーバ兼用型）

①サーバ02（PRIMERGY ES280）

ハードスベック
 CPU Pentium III 800Mhz
 MEMORY 640MB
 PageFile 初期 768MB 最大 1280MB
 HDD 9GB(C:4GB、D:5GB)
 OS WindowsNT4.0 SP6a
 アプリケーションサーバ
 INTERSTAGEV3.0L1.0(富士通)
 データベース
 Oracle8i WorkgroupServer 8.1.6(オラクル)
 Webサーバ
 Information Provider Pro(富士通)
 Java
 JSDK2.0(Sun)
 JDK1.2.2(Sun)
 帳票出力アプリケーション
 SuperVisualFormade Ver2.13e(霞システム)
 Javaクラスライブラリ Ver1.00a

クライアント数/台	応答時間/ミリ秒※1	CPU使用率/%※2	発生エラー合計/個※3
50	191691.0/6314.5/10.0	99.61	0
45	89018.0/5372.5/5.0	99.784	1
40	80315.5/5485.0/20.0	99.909	0
35	61618.5/4912.5/10.0	99.599	0
30	68699.0/4584.3/13.3	99.717	0
25	73676.0/3884.5/15.0	99.616	0
20	47768.5/2855.0/10.0	99.672	0
15	19718.5/1981.5/10.0	98.41	0
10	8162.0/995.0/10.0	83.582	0
5	3800.5/679.0/20.0	58.62	0

※1 最大/平均/最小 正確に計測できない場合は省略

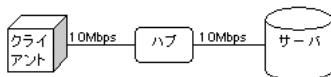
※2 WinNTの「パフォーマンスモニタ」使用 300秒のテスト前後10秒は誤差として計算外とした

※3 テスト使用プログラムでのエラー数の合計値 基本的には平均値(小数点以下四捨五入)

②クライアント01 (Fujitsu FMV-5233NU/W)

ハードスベック
 CPU MMXPentium 233Mhz
 MEMORY 64MB
 OS WindowsNT4.0 SP5

③ネットワーク環境



負荷テスト処理内容

- ①コードメンテナンス更新処理 1件 40%
- ②収発入力業務 表示(検索)処理 1/100000件 30%
- ③収発入力業務 起案書出力(プレビュー)処理 1件 30%
- PDF形式ファイル 出力サイズ:12KB
- 使用スクリプトファイル xxx1.scr, xxx1.dst, xxx1.cfm

表3 負荷テスト（物理3層その1）

実施日：平成12年8月23日～24日
 タイプ3（3層：Webサーバ・DBサーバ別型）

Webサーバ：DELL
 DBサーバ：PRIMERGY
 クライアント：FMV-5233

クライアント数/台	応答時間/ミリ秒※1	CPU使用率/%※2	発生エラー合計/個※3
50	194495.0/6020.0/15.0	67.899(Web)/51.603(DB)	1
45	129721.5/5370.5/25.0	67.796/48.971	0
40	111585.5/5517.5/25.0	67.102/50.725	0
35	103393.5/5349.5/30.0	67.8/50.367	0
30	71342.5/4487.0/30.0	66.981/48.89	0
25	64657.5/3756.5/20.0	67.433/49.621	0
20	56035.0/4185.0/15.0	66.977/50.9	0
15	22277.5/1892.0/30.0	67.062/48.079	0
10	19228.0/933.5/25.0	55.21/39.09	0
5	3039.5/408.5/25.0	29.923/28.128	0

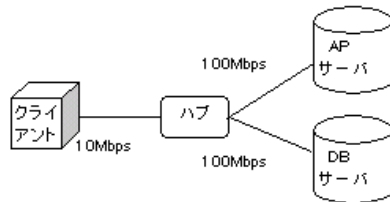
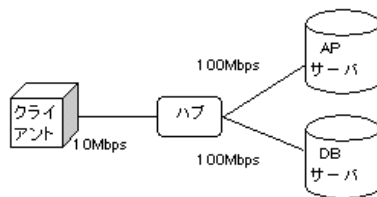


表4 負荷テスト（物理3層その2）

実施日：平成12年8月23日～24日
 タイプ3（3層：Webサーバ・DBサーバ別型）

Webサーバ：PRIMERGY
 DBサーバ：DELL
 クライアント：FMV-5233

クライアント数/台	応答時間/ミリ秒※1	CPU使用率/%※2	発生エラー合計/個※3
50	181271.0/7220.5/10.0	42.101(Web)/81.52(DB)	4
45	112957.5/5723.5/0	41.046/82.161	3
40	178216.5/6272.0/5.0	41.219/81.992	1
35	85818.5/5878.0/10.0	42.03/80.604	1
30	66420.5/5404.0/20.0	41.901/81.684	0
25	92733.5/5195.0/15.0	44.398/79.475	0
20	56035.0/4185.0/15.0	41.282/82.705	0
15	33043.0/2979.5/20.0	43.27/80.34	0
10	12988.5/1621.0/15.0	28.448/74.982	0
5	5423.0/903.0/20.0	28.016/51.521	0



脚注

- ⁱ ファイリングシステムとは、「ファイル法」のことであり、文書を必要に応じて迅速に利用可能に整理・保管する方法のこと。文書管理の改善の方法としては標準的である。一般的に、「簿冊方式」と「フォルダ方式」の運用に大別される。
- ⁱⁱ Java Database Connectivityの略で、Javaアプリケーションからデータベースを操作するAPIのこと。JDKのコアAPIとしてjava.sqlパッケージに実装されている。

-
- ⁱⁱⁱ Enterprise Java Beans. Java で記述したビジネスロジックのコンポーネントの総称. サーバサイドで動作する. Java2 以降に対応.
- ^{iv} Common Object Request Broker Architecture の略. 米国のオブジェクト技術標準化団体 OMG が策定している, 分散オブジェクト環境におけるオブジェクト同士のインタフェース仕様.
UNIX や Windows などプラットフォーム, C/C++ や Java などの開発言語に依存せず, 多くの開発システムで利用できる標準アーキテクチャ.
- ^v Dynamic Shared Object の略. 必要な時にだけ特定の機能拡張を行うことが可能になった. この拡張機能は「モジュール」と呼ばれる形式で提供されている. 現在組み込まれている DSO モジュールの一覧は, Apache の設定ファイル「httpd.conf」を参照すればよい.