

第1章

サーバ・ベース・コンピューティング (SBC)のメリット

ユーザインターフェースとしてのWebとWindows
SBCの仕組み / SBCのWANはなぜ速いか / SBCのメリットの体系
「WANでもサーバからのレスポンスが速い」がもたらすメリット
「クライアント用のアプリケーションをサーバで実行する」がもたらすメリット
ターミナルサービスの導入で失うこと

1.1

ユーザインターフェースとしての WebとWindows

レコード単位でのインタラクティブ性

Web画面は紙芝居です。ページ単位で画面が変わります。仮に、表計算ソフトがWeb化されたとしましょう。ひとつのセルへの入力のためにページ全体が更新されるようになっていていると思います。

Windowsはオセロゲームです。ひとつの入力に対しレコード単位で反応があります。例えば、表計算ソフトでは、レコード単位（セル単位）で入力すると、連関する計算式があるセルだけが更新されます。つまり、Windowsインターフェースはレコード単位でのインタラクティブ性に優れているのです。

表計算ソフトのように、業務システムの多くはレコード単位での入出力を必要とします。大量の伝票入力がある会計システムをWeb化してしまった企業があります。伝票入力のために多くの派遣社員の受け入れが必要になってしまい、かえって経費が増えてしまったそうです。会

図1 レコード単位でのインタラクティブ機能の例

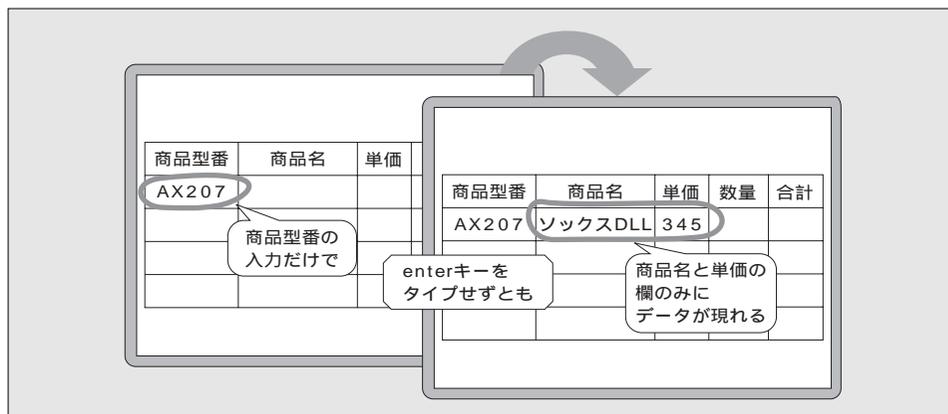
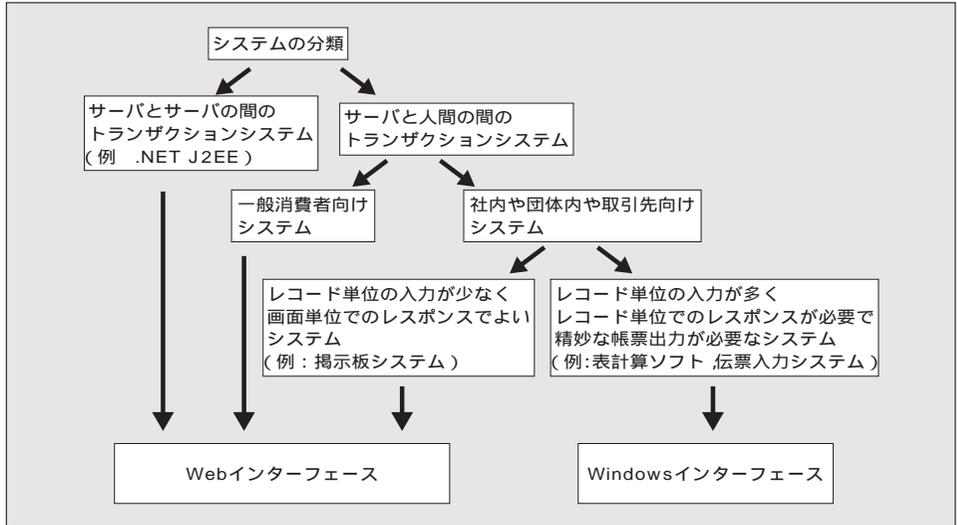


図2 WebとWindowsインターフェースの得意なシステム分野



計システムを刷新した結果として経費が増えてしまったのは、レコード単位でのインタラクティブ性を過小評価したことによる失敗です。

もうひとつ、Web画面は、印刷用帳票のピッチ合わせが困難です。ブラウザの種類やバージョンの違いにより幅が変わったりします。Windowsでは破線、二点破線、太線、二重線、など多様な線が使え、かつそのピッチ幅は固定できます。帳票印刷をとまなう業務ソフトでは、Web画面でなくWindows画面にせざるをえません。

Webに向いている業務

しかし、レコード単位での入力が少なく、出力がページ単位でよいというシステムもあります。掲示板システムなどのグループウェアがその例です。このようなシステムにはWeb画面が向いています。また、クライアント側のライセンス料を安あがりにする一般消費者あてのB2CシステムにもWebがよく使われます。画面あたりの入力レコード項目が多くても、レスポンスが画面単位となって遅くてもよいというシステムです。B2Cシステムでは、企業内の伝票入力システムほど大量の入力はしませんから。

人間とサーバとの間のやりとりではなく、サーバとサーバとの間のやりとりにも、Webは向いています。「在庫数の問い合わせコマンド」などをXML形式で生成すると、HTTPプロトコルでファイヤウォールを通過して相手先企業のサーバへ送ることができます。これにより相手先企業のサーバと自社のサーバ、これらのサーバ同士でデータのやりとりができるようになります。Webサービスがその例です。.NETやJ2EEなどの技術がカバーしようとする分野がこれです。Webサービスを導入してもユーザインターフェースにはWindows GUIが使われるでしょう。インターフェースとしてのWindowsとWebの得意な分野を図2に整理しました。

企業の基幹業務の多くは、レコード単位での入力が多く、レコード単位でのレスポンスの良さが要求されます。また野線の多い帳票出力も必要です。したがって、企業の基幹業務でのユーザインターフェースは、WebでなくWindowsを採用せざるをえません。

サーバ・ベース・コンピューティング

ところが、Windowsインターフェースのシステムには重大な弱点があります。それは、アプ

リケーションをクライアントのパソコンで実行することと、サーバを地理的に分散配置することです。いわゆるクライアント/サーバ・システムの問題点です。

Webコンピューティングでは、アプリケーションはサーバでのみ実行し、サーバを地理的に集中して設置することも可能です。したがって、Webコンピューティングはシステムの運用と管理の面でクライアント/サーバ・システムより優れています。システムの安定性の面でも優れています。各クライアントでアプリケーションを実行するクライアント/サーバ・システムでは、どこかのパソコンのアプリケーションが毎日のようにハングアップしているからです。

そこで、Windowsインターフェースのアプリケーションをサーバで集中実行するようにしようというのが、これから説明するサーバ・ベース・コンピューティング(SBC)です。本書は、コンピュータシステムの運用や管理や構築にたずさわる多くの人に、「サーバ・ベース・コンピューティングをどのように実現するか」と「サーバ・ベース・コンピューティングへの移行に伴う問題点をどのように克服するか」を理解していただくことを目標としています。

Webコンピューティングとサーバ・ベース・コンピューティングは、図2のように得意な領域が異なります。したがって、今後もこの2つのコンピュータ・アーキテクチャは、併存していくことになるでしょう。確かなことは、将来的にはクライアント/サーバ・システムからの脱出が避けられないということです(図3)。

移行の容易さ

サーバ・ベース・コンピューティングでは、いま使っているクライアント用のアプリケーションをそのままサーバにインストールし、サーバで実行します。4.1.2項「アプリケーションのマルチユーザ対応」(p.215)の記述を考慮する必要がありますが、基本的には、いま使っているアプリケーション資源はそのまま継承できます。これが、移行に関してWebコンピューティ

図3 Webコンピューティングとサーバ・ベース・コンピューティングのインタラクティブ性とアプリケーションの実行場所

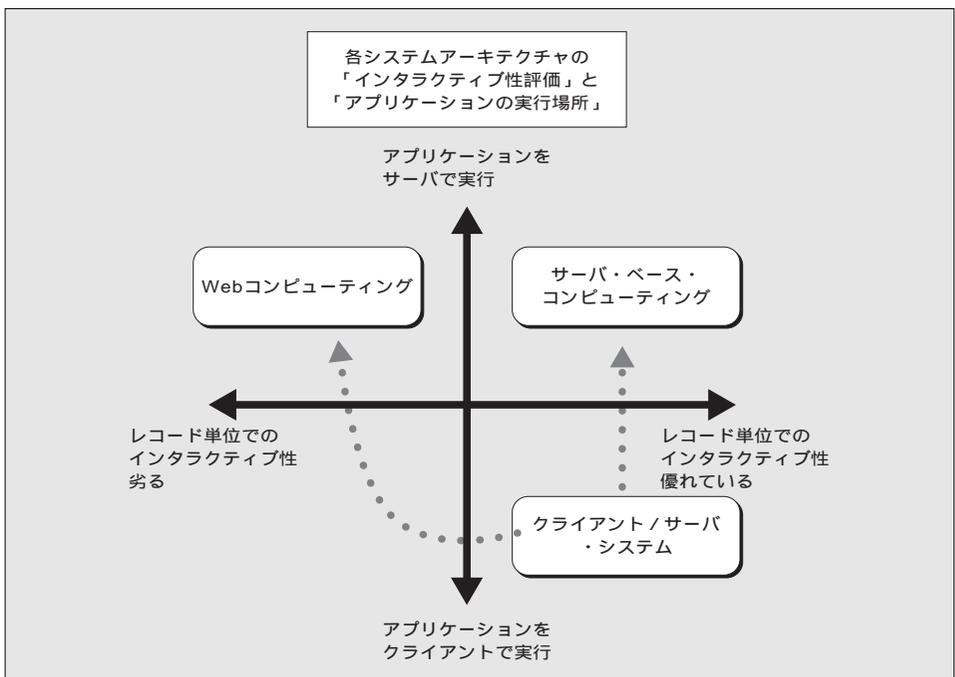
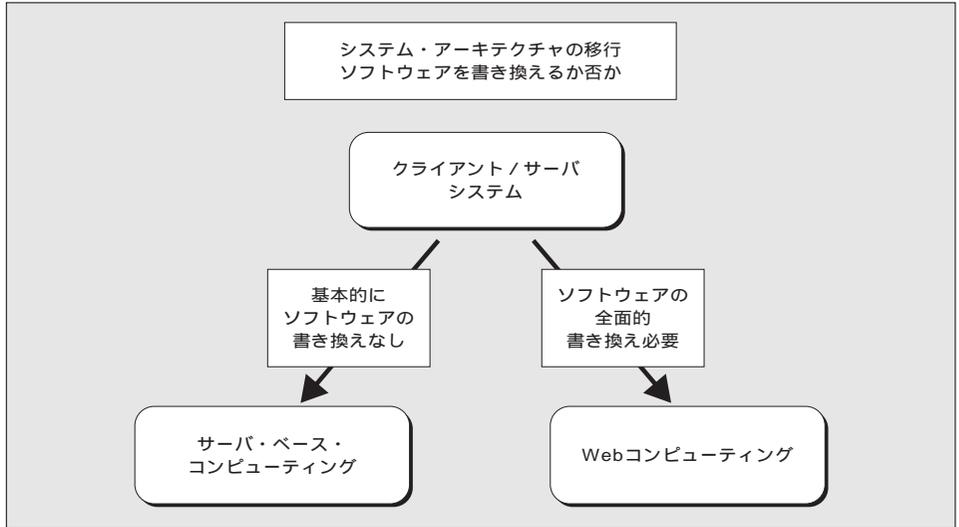


図4 サーバ・ベース・コンピューティングではクライアントのソフトウェアを書き換えなしで移行できる



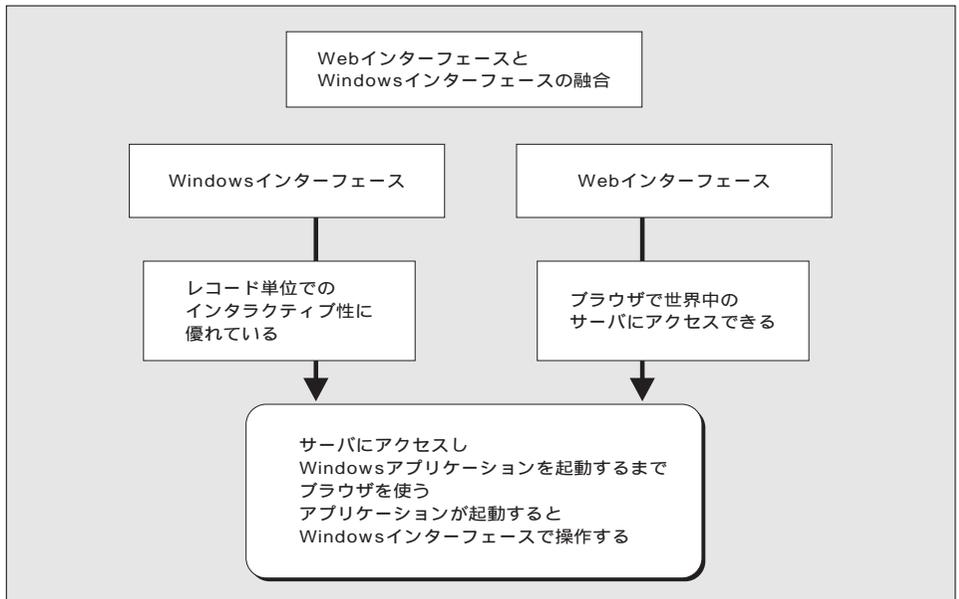
ングよりサーバ・ベース・コンピューティングのほうが有利なもうひとつのメリットです(図4)。

WindowsアプリケーションとWebの融合

しかし、Webコンピューティングで使うブラウザは便利なものです。あらためてブラウザの長所を下記します。

* **サーバの指定** URLを入力するだけでサーバを指定できます。Windowsのエクスプローラのネットワークコンピュータで目的のサーバを指定するのに比べ、はるかに汎用的な指定方法です。アクセスが許されている世界中のサーバにアクセスできます。

図5 MetaFrameをアドオンすればWebとWindowsのGUIが融合できる



* **誰でも使える** パソコンを使う人のほとんどがブラウザを使えます。URLとIDとパスワードを教えれば、誰でも指定のサーバへアクセスできます。

* **どのパソコンにもインストールされている** ブラウザがインストールされていないパソコンはまずないと思われます。

いままで説明の便宜上、Webコンピューティングとサーバ・ベース・コンピューティングを、対立するアーキテクチャとしてとらえてきました。しかし、MetaFrameをアドオンしたサーバ・ベース・コンピューティングではブラウザを使って、Windowsアプリケーションがインストールされたサーバにアクセスできます。これにより、Webコンピューティングとサーバ・ベース・コンピューティングが融合されます（p.42の2.3.C1項「ブラウザでサーバのWindowsアプリケーションへアクセス」を参照）。

1.2

サーバ・ベース・コンピューティングの仕組み

クライアント/サーバ・システムでは、おもにクライアント側にアプリケーションを搭載し、そのアプリケーションとデータベース・サーバとの間で、データやSQLなどのコマンドをやりとりしていました。

一方、サーバ・ベース・コンピューティングでは、クライアントで使っているアプリケーションをWindows2000サーバに搭載します。現在使っているWindowsクライアントのアプリケーションは基本的にはそのままWindows2000サーバで実行できます。そして、Windows2000のターミナルサービス機能を使って複数のクライアントでそのアプリケーションを共用します。これでWindows2000サーバがアプリケーション・サーバになります。このWindows2000サーバには通常MetaFrameをアドオンします（図6）。

アプリケーション・サーバに搭載したアプリケーションはデータベース・サーバと同様の手

図6 クライアント/サーバ・システムからサーバ・ベース・コンピューティングへの移行

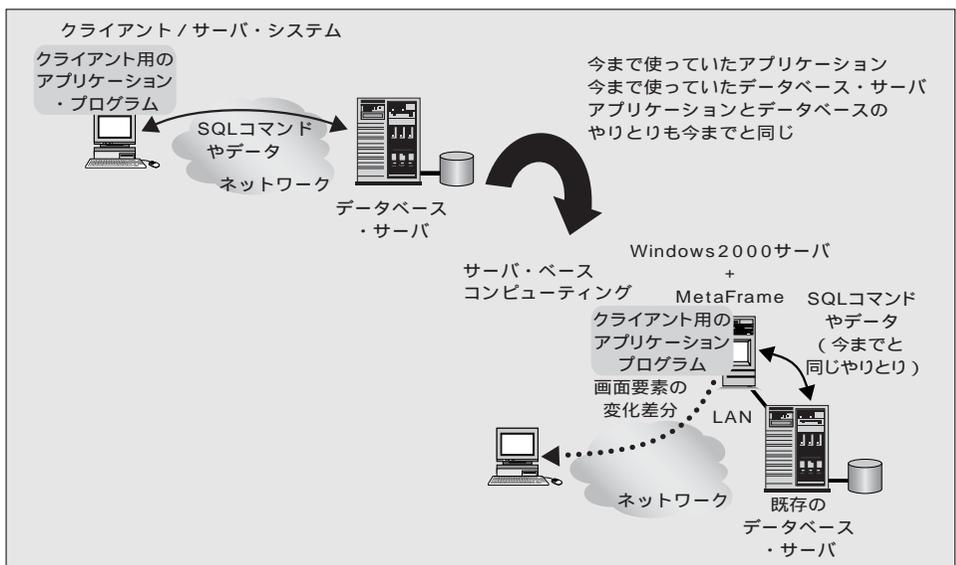
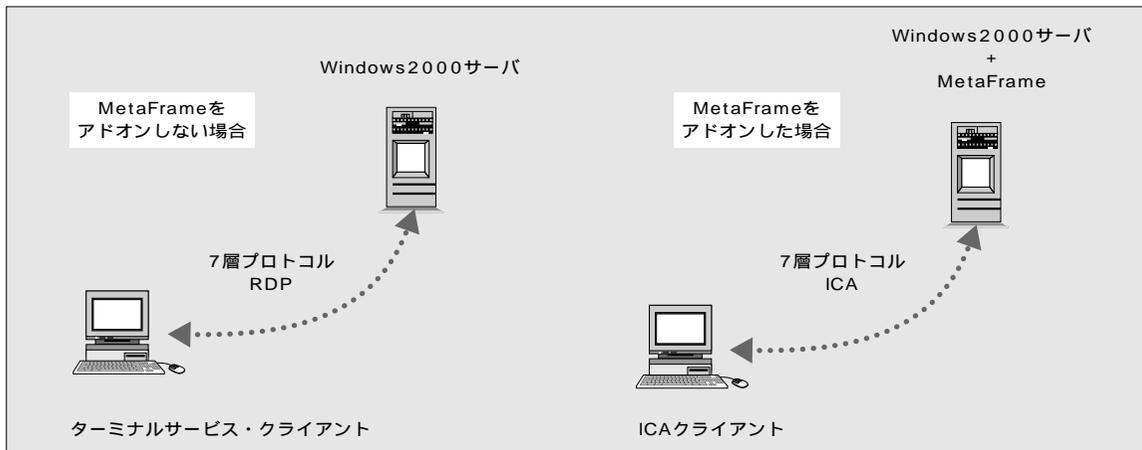


図7 MetaFrameをアドオンした場合とアドオンしない場合の違い



順でデータやSQLなどのコマンドのやりとりを行います。データベース・サーバは、ハードウェア的にもソフトウェア的にも変更する必要はありません。もちろんUNIXをデータベース・サーバOSとして使い続けることも可能です。

サーバ・ベース・コンピューティングでは、アプリケーション・サーバからクライアントの方向には、画面の画素のみが流れます。流れる画素はアプリケーションのプロセスを反映しての画面変化部分のみです。

MetaFrameをサーバにアドオンしない場合には、クライアントとサーバの間の7層プロトコルはRDP (Remote Desktop Protocol) というプロトコルを使います (図7)。RDPは、国際電気通信連合が定義したプロトコル標準のT.120ファミリに準拠したマイクロソフト独自のプロトコルです。また、クライアントに「ターミナルサービス・クライアント」という名のソフトモジュールをインストールします。サーバから送られてきた画素ビットをディスプレイに表示することが「ターミナルサービス・クライアント」の役割の一つです。

MetaFrameをアプリケーション・サーバにアドオンした場合、クライアントとサーバの間の7層プロトコルはICA (Independent Computing Architecture) というシトリックス独自のプロトコルを使います。ただしMetaFrame XP FR1では、ICAをhttpsプロトコルの中にカプセル化してSSL手順で送ることもできます (p.49の2.4.D1を参照)。

クライアント側には、「ICAクライアント」という名のソフトをインストールします。「ICAクライアント」の役割のひとつは、「ターミナルサービス・クライアント」と同様、サーバから送られてきた画素ビットをディスプレイに表示することですが、それ以上に多くの機能をもっています。

ICAは、RDPより狭帯域回線では、レスポンスがかなり速くなることが実証されています。これが、MetaFrameを導入する理由のひとつとなっています。p.251の4.4.3.1項「通信回線帯域の考慮」を参照ください。

本書は、クライアント用のアプリケーションをWindows2000サーバで実行することを前提に記述しています。しかし、Windows2000の前身である、NT/TSE (Microsoft Windows NT, Terminal Service Edition) をサーバOSとして使っても、サーバ・ベース・コンピューティングは実現できます。Windows2000サーバには、下記の3種類があります。

Microsoft Windows2000 Server
Microsoft Windows2000 Advanced Server
Microsoft Windows2000 Data Center

このどれを使っても、サーバ・ベース・コンピューティングは実現できます。MetaFrameをアドオンする場合は、この3種類のどれを使っても、機能や性能やスケーラビリティに変わりはありません。Windows2000 Serverの後継であるWindows .NET Serverでもサーバ・ベース・コンピューティングは実現できます。

1.3

サーバ・ベース・コンピューティングのWANはなぜ速いか

クライアント/サーバ・システムでは、クライアント側のアプリケーションとデータベース・サーバとの間で、データや、SQLなど様々なコマンドをやりとりします。それもひとつの画面が表示されるまで何回もデータやコマンドがやりとりされます。

本来、クライアント/サーバ・システムは、クライアントとサーバとの間を高速の回線、それも通常は10Mbps以上のLANで接続することを想定して考案されたコンピューティング方式であり、LANが普及する前にはクライアント/サーバ・システムは存在しませんでした。LANですと、クライアントでのディスプレイ画面が更新するまでそれほどの時間はかかりません。ところが、LANに比べてはるかに遅い通信回線経由でクライアント/サーバ・システムを実行するとディスプレイ画面の更新にひどく時間がかかります(図8)。

サーバ・ベース・コンピューティングではクライアントの仕事を代行するWindows2000サーバとデータベース・サーバとの間は、同じデータセンター内ですからLANでつながります。アプリケーション・サーバからクライアントの方向には、画面画素の変化差分のみが送られます。その結果、サーバ・ベース・コンピューティングでは、サーバからのレスポンスが驚くほど速くなります(図9)。

画像を使わず、色数も16色でよい一般の業務アプリケーションなら、サーバ・ベース・コンピューティングでは64kbps回線でも、LANなみのレスポンス速度が得られます。「64kbpsの通信回線経由でクライアント/サーバ・システムで運営していたときはサーバからのレスポンス

図8 クライアント/サーバ・システムをWANで組むとひどく遅くなる

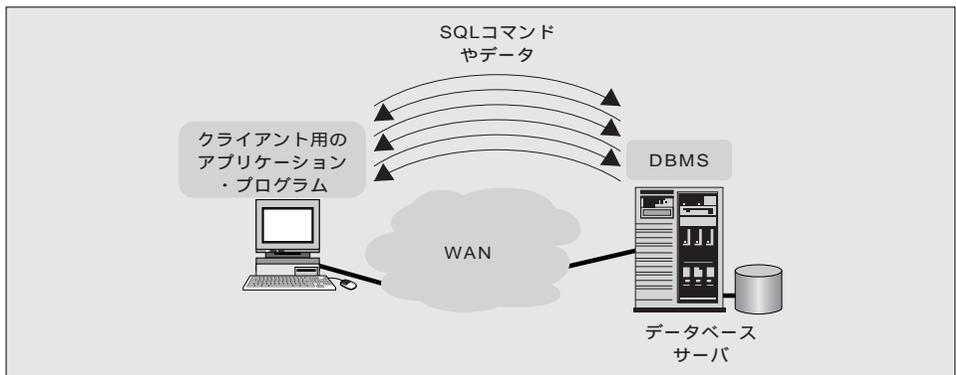
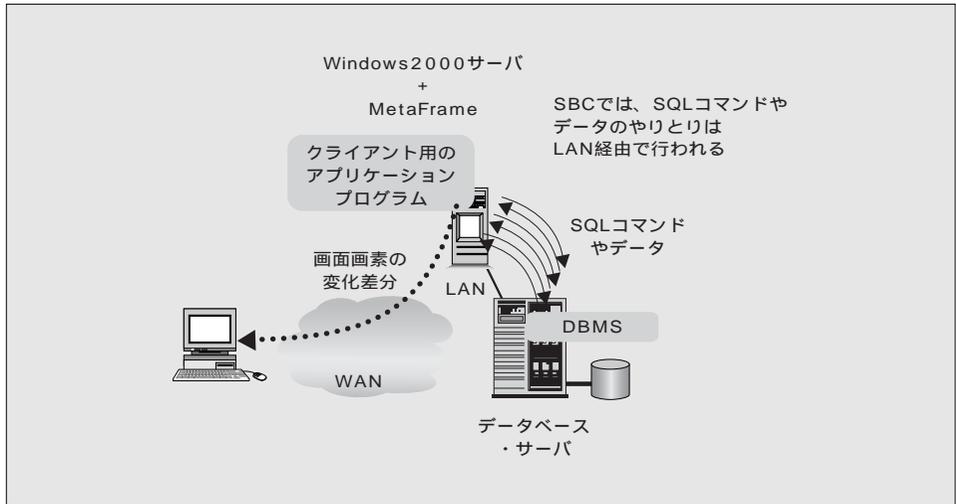


図9 サーバ・ベース・コンピューティング (SBC) では、アプリケーションの実行結果の画面の変化差分の画面のみをクライアントに出力する



に90秒もかかっていたが、サーバ・ベース・コンピューティングに移行したら6秒に縮まった」という報告もたくさんあります。

1.4

サーバ・ベース・コンピューティングの メリットの体系

WANでもサーバからのレスポンスが速い

クライアント用のアプリケーションをサーバで実行する

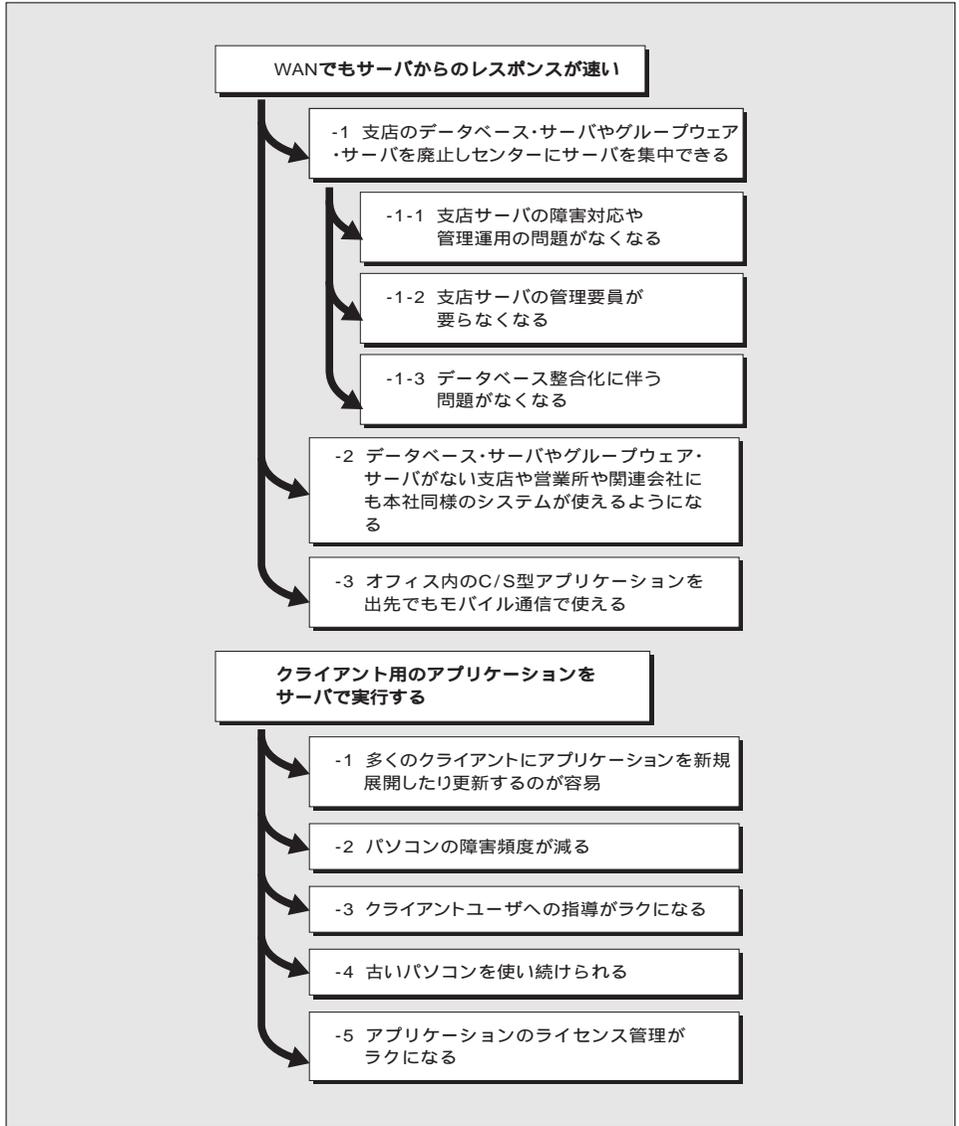
サーバ・ベース・コンピューティングの基本的なメリットはこの2つです。この2つの基本的なメリットが図10のように多くの派生的メリットを産みます。

これらサーバ・ベース・コンピューティングのメリットは、クライアント/サーバ・システム

「いまは64kbpsのフレームリレーだけど、ADSLなどの導入でブロードバンド時代になるからサーバ・ベース・コンピューティングのソリューションは暫定的だ」と、評論する人がいます。確かに、LANなみの速度の通信回線を用意すれば、クライアント/サーバ・システムのままでもサーバの地理的集中によるレスポンス問題は、理論的には解決できます。しかし、これはITの構造改革を避け、現状を維持したい願望でしかありません。既存のクライアント/サーバ・システムのまま、通信回線を高速にして解決しようとする方法では、図10の「クライアント用のアプリケーションをサーバで実行」がもたらすメリットを享受できません。また、クライアント/サーバ・システムのまま、「ADSL」+「通信パイプとしてのインターネット」を利用する方策は、Firewallに非標準のポートを開けざるをえないことや暗号化などのセキュリティ上の問題も控えています。セキュリティ問題は、MetaFrame XP FR1をWindows2000サーバにアドオンすると、SSL (Secure Sockets Layer) で安全に通信ができるようになることで解決します。

それやこれやで、サーバ・ベース・コンピューティングのほうが逆に、「ADSL」+「通信パイプとしてのインターネット」の料金が安いというメリットを享受できるようになります。

図10 サーバ・ベース・コンピューティングのメリット



ムの特長の裏返しともいえます。つまり、クライアント/サーバ・システムでの本質的な問題点がサーバ・ベース・コンピューティングに移行することで解決できるわけです。

1.5

「WANでもサーバからのレスポンスが速い」 がもたらすメリット

「90秒のレスポンスが64kbps回線でも6秒に縮まった」という事例は、サーバ・ベース・コンピューティング（SBC）の根源的なメリットです。このメリットを応用すると下記のような展望が開けてきます。

支店のサーバを撤廃できる