

クライアント/サーバシステムによる分散システムの構築

3B-3

松尾 典子

(株) マイカルシステムズ

(共同研究者：富士通ラージシステム研究会平成5年度クライアント

サーバシステムによる分散システムの構築I分科会)

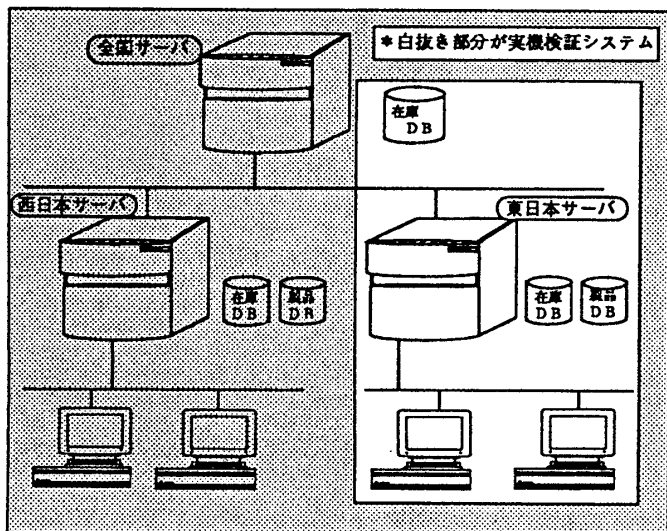
1. はじめに

ここ数年、「クライアント/サーバシステム」を採用する企業・団体が増えているが、これまでホストで稼働してきた基幹業務をそのままダウンサイジングしている例はまだまだ少ない。高度なトランザクション処理が要求される業務系システムで避けることができない課題として挙げられるのが更新処理であると言える。そこで、演者ら研究会では基幹系業務（業務系システム）でメインとなる更新処理での実機による検証に重点をおいた。既存のホストシステムからの単なるダウンサイジングではなく、クライアント/サーバシステムにおいてクライアント処理型の「パソコン上のアプリケーションとRDBを連携した更新業務」が実業務に適用できるかを表計算連携ソフトでの開発と4GLでの開発に分けて比較検証し、クライアント/サーバにおける構築のポイントを挙げることにする。

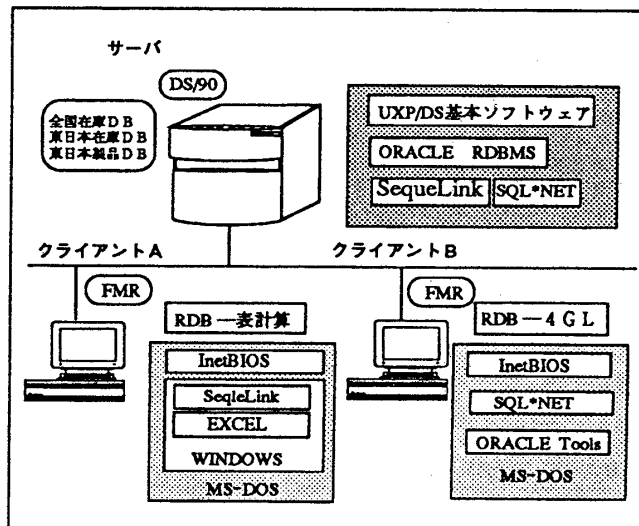
2. システム概要

対象業務はクライアント/サーバシステムが従来の基幹システムにどこまで近付けるかを見極めるために更新トランザクション処理の多い「在庫管理システム」を選択した。このシステムの構築にあたっては全国各地の支社・支店・営業所・工場を結ぶことによって在庫管理を実現する実用性、また広域通信網の利用などの今後の拡張性等も考慮している。

想定システム構成概要



検証システムハード・ソフト構成



3. 検証内容と評価

実機検証システムのクライアントA（表計算ソフト）とクライアントB（4GL）を使用し、開発面及び操作面について比較検証を行った結果を以下の表に示す。

Construction of a Distribution System based on Client/Server System Methodology

Noriko Matsuo

MYCAL SYSTEMS INC.

| | | RDB-表計算 | RDB-4GL |
|-------|-------|--|---|
| 操作性 | 出力画面 | GUI ○ 文字の大きさ、ボタンの設定は自由 ○ 全体の見栄えは非常によい | CUI × 文字の大きさは固定 × 表計算より見栄えする |
| | 操作方法 | × 次項目への自動移動が困難 ○ マウスによるランダムな項目選択 × ファンクションキーは標準設定では使用不可 | ○ 次項目への自動移動ができる × マウスによるランダムな項目選択は不可 ○ ファンクションキーの使用可能 |
| レスポンス | 検索処理 | ○ 1.39秒 (1万件より1件検索し表示した場合) | × 6.10秒 (1万件より1件検索し表示した場合) |
| | 更新処理 | × 34.91秒 (100件中10件を更新した場合) | ○ 8.50秒 (100件中10件を更新した場合) |
| コスト | 開発工数 | ○ 25時間 | ○ 24時間 |
| | ソフト費用 | × 約280万 [クライアント: SequeLink, Windows, Excel] [サーバ: SequeLink, ORACLE] | ○ 約270万 [クライアント: SQL*Net, Tools] [サーバ: SQL*Net, ORACLE] |

* 今回の検証は時間の制約上、インデックスなしチューニングなしで行ったものである。

レスポンスに関してはアプリケーションの違いにより差が出ている。

* ○印は比較的良好

×印は比較的不善

4. クライアント/サーバシステム導入における構築のポイント

今回の実機検証によって得られた構築のポイントを述べる。

(1) 機能、性能だけでソフトを選択するのは危険である。

選択にあたっては、レスポンス特性、アプリケーション構築の簡便さ・柔軟性は重要であるが、マニュアルの充実度、ベンダのサポート体制、製品群の組み合わせに実績があるか、また、その組み合わせによる大規模システムの事例があるかが問われる。

(2) クライアント/サーバシステム導入にあたっては事前準備が重要である。

今回、各メンバにUNIX, RDB, LAN, PCなどの知識にばらつきがあったため必要以上の時間を要したが、クライアント/サーバシステムとは全く新しい技術であるという意識が必要であり、新技術の情報収集・習得、あるいは要員の確保などの事前準備をしておかなければならない。

(3) 表計算ソフト、4GLでの大規模システム構築/分散開発は困難である。

表計算ソフトと4GLはイベントトリガ型の開発が可能のため、従来のホスト開発と比べて高い生産性を得ることができた。しかし、大規模システムを開発するためにはコーディング規約の標準化、モジュールの共有化を考慮しなければならない。また、レスポンスの向上を計るためにはSQL文の内部ロジックを理解し、SQL文の発行回数を少なくすることなどが必要である。

(4) パソコンのインストールは予想以上に大変である。

現状ではインストールおよび環境設定については個々の製品のインストールマニュアルを体系化した独自のマニュアルを作成しておく必要がある。今後はメモリの設定、ドライバの設定など環境の違いを意識しないような簡略化したソフトが望まれる。

(5) 障害はベンダに問い合わせで容易に解決しない

レスポンス悪化において、原因がサーバかクライアントかネットワークか切り分けに手間取ったが、各市販ソフトに内部処理のシーケンスやログを出力する機能が実装されれば問題の切り分けも容易になると思われる。ユーザ側においては自社での教育体制の強化は必須である。

5. おわりに

「クライアント/サーバシステム」は本当に使えるのかという疑問のもとに集まったメンバであるが、実機開発を通して、ツール類の使い勝手、開発のテクニックなど机上では得られない貴重なスキルを習得できたと言える。今後は、更に実際に分散されたシステムの運用・管理などに着目していきたい。