

簡易言語による大規模分散型システム構築環境VGUIDEの適用

川崎隆二 黒川裕彦 山本修一郎
NTT ソフトウェア研究所

VGUIDE (Visual and General User Interface for Database Environment)は、クライアントサーバ型情報システムの開発・実行環境である。VGUIDEは以下の特徴を持つ。

- (1)データベース操作簡易言語により、システム開発の生産性が高い
- (2)高負荷な大規模システムにも適用できるリアルタイム制御機能により、スケーラビリティが高い
- (3)簡易言語の実行環境が多様なベンダ環境で用意されており、アプリケーションの移植性が高い

本論文は、分散型情報システムの構築上の課題として、ミドルウェア環境の統合、クライアント/サーバモデルの統合、アプリケーション制御形態の統合が必要であることを明らかにする。次に、これらの課題を解決するためのVGUIDEの構成と機能並びに、その効果について述べる。最後に、VGUIDEの適用事例を紹介する。

VGUIDE:

4GL Application Platform for Large Distributed Information System

Ryuji Kawasaki, Hirohiko Kurokawa, Syuichiroh Yamamoto

NTT Software Laboratories

VGUIDE (Visual and General User Interface for Database Environment) is the integrated environment for developing and running distributed information systems based on the client server model. Its main features are as follows.

- (1) Its 4th generation language for developing database applications reduces system development cost.
- (2) Its realtime control monitor has the scalability from small systems to heavy loaded large systems.
- (3) 4GL application programs can be ported among multi vendor machines and operating systems

First, we discuss technical problems for developing distributed information systems. To address these problems, it is necessary to integrate middleware environments, client server models and application control mechanisms. Next, we describe the configuration, functions and effectiveness of VGUIDE. Also, we describe an actual example with VGUIDE.

1. はじめに

VGUIDEは、クライアント/サーバ型情報システムの開発・実行環境である。

本論文では、2章で分散型情報システムの構築上の課題として、(1) ミドルウェア環境の統合が必要であること、(2) クライアント/サーバモデルの統合が必要であること、(3) アプリケーシ

ョン制御形態の統合が必要であること、(4) リモートSQLの高速化が必要であること、を明らかにする。3章では、これらの課題の解決を目的とするVGUIDEの構成を述べ、4章では、VGUIDEの機能について述べる。5章ではVGUIDEの効果について定量的に述べる。6章では、実際の適用事例を紹介する。7章では、

VGUIDEと関連技術を比較する。最後に8章では、まとめと今後の課題を述べる。

2. 分散型情報システム構築上の課題

本格的な情報システムをクライアントサーバ型分散情報システムの形態で構築するには、以下の4つの課題がある。

2.1 ミドルウェア環境の統合

今日、クライアント／サーバ型情報システムの構築を支援するミドルウェアとして、中小規模システムを対象に、簡易言語でプログラミングが可能な環境を持つミドルウェア（表示層やデータベース操作の定義をビジュアルな操作で可能とし、その実行コードを生成する）が提供され、システム開発の生産性向上に大きな効果を上げている。一方、大規模な基幹業務システム向けのミドルウェアとしては、TPモニタが用意され、大規模システム構築に必要なトランザクション制御、オンラインリアルタイム制御、障害時のリカバリ制御等の機能を提供している。

しかし、簡易言語の利用できるプログラミング環境で、数百クライアントを超える大規模システムを実用的な性能で実現できるミドルウェアは極めて少ない。本格的な情報システムの構築を容易にするには、TPモニタの持つ高性能・高信頼性と簡易言語の持つ高生産性を兼ね備えた環境が必要である。

2.2 クライアント／サーバモデルの統合

図1はクライアントとサーバへの機能配置に着目したクライアント／サーバモデルの分類であり、以下の4つのタイプが考えられる。

(1) 遠隔表示型：業務を実現する機能層のアプリケーションを全てサーバに搭載し、クライアントへの画面表示にはウィンドウシステムの仕組みを利用する形態。システムの形態としては集中型に近い。

(2) 遠隔DB型：機能層のアプリケーションを全てクライアントに搭載し、サーバマシンをDBサーバとして利用する形態。クライアントにローカルなデータを持つ場合もある。

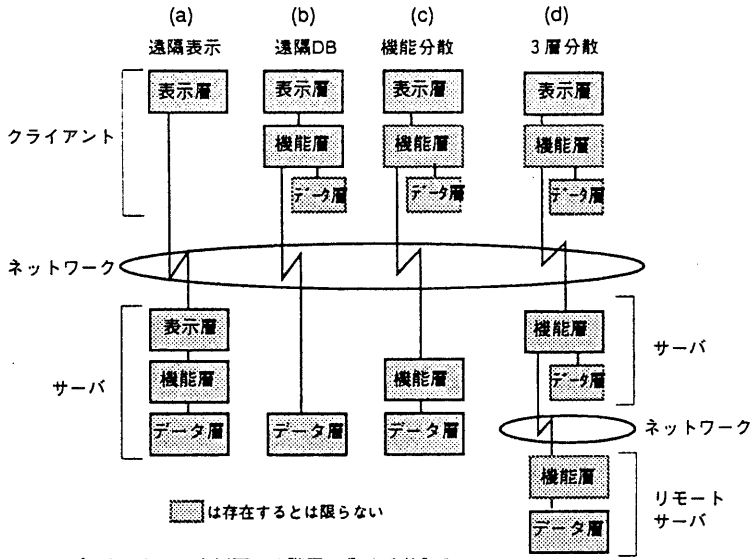
(3) 機能分散型：機能層のアプリケーションをサーバに搭載し、クライアントのアプリケーションから呼出す形態。クライアントの表示層アプリケーションからサーバのアプリケーションを呼出す場合と、クライアントにも機能層アプリケーションを持ちこの機能層アプリケーションからサーバのアプリケーションを呼出す形態がある。(1)、(2)と比較して柔軟で高性能なシステムを構築し易い。

(4) 3層分散：(3)の機能分散型に加えて、データ構造やDBMSからの独立性を高めるために、機能層とデータ層を明確に分離すると共に、アプリケーションやデータの配置をアプリケーションから隠蔽することをねらった形態。

個々のシステムでこれらの形態のいずれを選択するかは、システムに要求される条件に依存する。このため、情報システムを柔軟に構築可能とするには、図1(a)～(d)の形態がいずれも構築可能である必要がある。

例えば、システムを簡便に構築したい場合には、プレゼンテーション層とファンクション層の分離を考慮する必要がないために遠隔DB型を選ぶことが多い。また、非定型な業務のためのシステムでは、動的にSQLを組み立てられる必要があるため、同様にDBサーバとしてサーバマシンを利用する遠隔DB型となる。一方、高性能で柔軟なシステムを目的とするなら機能分散型、機能分散のメリットに加え、データ構造やデータの配置を隠蔽したいなら3層分散型が必要となる。

従来のクライアント／サーバシステム向け簡易言語は、「遠隔表示型」か「遠隔DB型」にしか適用できない。より高度な「機能分散型」「3層分散型」に対しては、簡易言語で利用できる



アプリケーションを以下の3階層に分けて考える
 表示系 (プレゼンテーション層) : ウィンドウを制御し、ユーザとの対話を行う部分
 機能系 (ファンクション層) : アプリケーションとして機能を組み込む部分
 データ層 : DBアクセス部分でDBMSとインタフェースを持つ部分

図1 クライアント/サーバモデルの分類

実用的な環境は整っていないため、現状では、遠隔の手続き呼び出しであるRPC (リモートプロシージャコール) が利用可能なC言語等を熟知した技術者によるシステム構築が必要という問題がある。

2.3 アプリケーション制御形態の統合

本格的な分散型情報システムでは、会話制御機能だけでなく、短時間での応答を保證すべきリアルタイム制御機能、月次処理等のためのバッチ制御機能が必要である。ところが、従来のシステムでは、これら3つの制御機能を持ち、簡易言語を利用できるアプリケーション開発・実行環境がないという問題があった。

従来の簡易言語の実行環境ではUNIX等の会話処理方式をそのまま利用しているため、端末毎にサーバ上でサービス主体となるプロセスが生成され、接続クライアント数に応じてサーバ側ではメモリ等の計算機資源を消費する。このため、一定のクライアント数を越えると処理時間が長大化し、サーバの処理能力の限界を生ずる

ため、リアルタイム型の処理には適用できなかった。

2.4 リモートSQLの性能向上

クライアントからのSQLによる遠隔DBへのアクセス方式では、ネットワークの負荷の影響を受けやすく負荷が高くなると十分な性能が得られないという問題があった。

3. VGUIDEのシステム構成

VGUIDEの目的は、2章で述べた分散情報システム構築の課題を解決する統合環境を実現することである。図2にVGUIDEのシステム構成を示す。

クライアントには、各種の市販GUIパッケージからVGUIDEのサーバ機能を利用可能とするための市販プログラム連携機構がある。クライアントアプリケーションは、VisualBASIC、PowerBuilder、SQLWindows、LotusNotes等で開発できる。また、クライアントOSとしては、Windows、OS/2、Macintoshが使用できる。

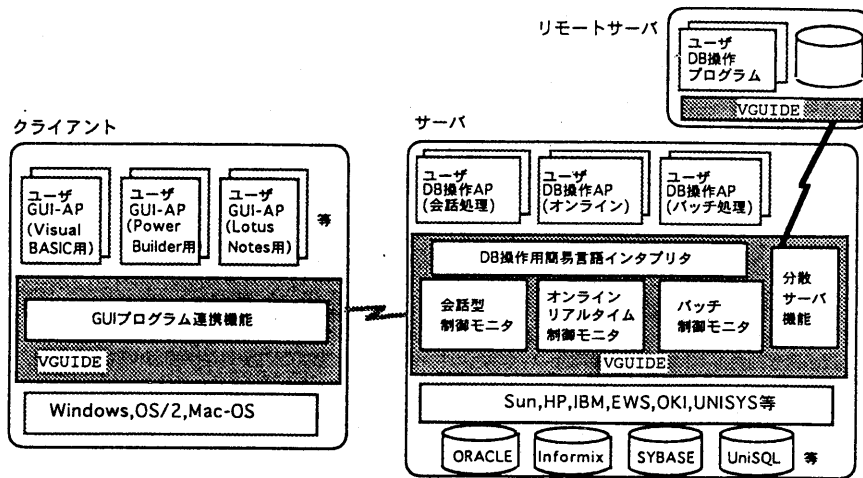


図2 VGUIDEのシステム構成

サーバには、簡易言語インタプリタ、モニタ、分散サーバ機構がある。簡易言語インタプリタは、DB操作簡易言語の解釈実行を行う。モニタには、会話制御モニタ、オンラインリアルタイム制御モニタ、バッチ制御モニタの3種類があり、これらのモニタの上で同一の簡易言語インタプリタが動作する。分散サーバ機構は、リモートサーバのアプリケーションの呼出し、リモートサーバのデータベースへのアクセスを行い、サーバ間でのアプリケーション分散、データベース分散を可能にする。サーバOSとしては各ベンダのUNIX-OSが使用できる。DBMSとしては、ORACLE、Informix、SYBASE、UniSQL等が利用できる。

表1 VGUIDE簡易言語のデータベース操作基本機能

分類	コマンド	機能
データベース定義	CREATE TABLE DROP TABLE ALTER TABLE	表の定義/削除/カラム追加
	CREATE VIEW DROP VIEW	ビューの定義/削除
	CREATE INDEX DROP INDEX	インデックスの定義/削除
	GRANT REVOKE	アクセス権限の定義/削除
	SELECT	条件に合致するレコードの検索
データベース操作	UPDATE	レコードの更新
	INSERT	新しいレコードの追加
	DELETE	レコードの削除
	LOCK TABLE SET LOCK MODE SET ISOLATION	ロックの制御
トランザクション制御	LUST	トランザクションの開始
	COMMIT ROLLBACK	トランザクションの終了
その他	TRANS DBLOAD	検索結果のファイル出力 ファイル入力によるデータベース更新
	EXEC CALL	簡易言語AP、3GL等の利用者AP、ストアードプロシージャの呼出し
	DWLD	簡易言語アプリケーションの作成した検索結果の取得

4. VGUIDEの機能

4.1 データベース操作基本機能

VGUIDEの簡易言語は、BASICを拡張したインタプリタ型の簡易言語である。この簡易言語では、表1に示すようにデータベース操作とトランザクション制御のためのSQLを記述できる。

4.2 データ入出力領域の隠蔽機能

C言語やCOBOLなどの3GLでDBアクセスを行

う場合には、SQLとプログラム言語間でデータを受け渡すための変数（ホスト変数）を、取り扱う全てのデータタイプに対して定義する必要がある。また、DBMSとプログラム言語間で制御情報を受け渡すためのシステム通信域の定義が必要になる。これに対して、VGUIDEの簡易言語では、これらの定義は不要であり、アプリケーションプログラムでのインタフェース定義を大幅に簡略化できる。

4. 3 AP制御構造の単純化機能

VGUIDEの簡易言語では、SELECT文での検索結果をアプリケーションプログラムがレコード毎にフェッチする処理を不要とし、SELECT文発行時にデータセットと呼ぶ検索結果格納用の2次元配列へ一括格納する。アプリケーションプログラムからの検索結果の参照はデータセットへのアクセス関数、および検索テーブル数の取得関数を用いて行う。これによって、アプリケーションプログラムの制御構造がデータ構造を反映して複雑になることを防止できる。

4. 4 入出力に関する定型処理の自動化機能

SQL発行毎に行うエラー判定とエラー処理の記述を不要とし、オンルーチンへ自動的に制御を移行する。また、入力メッセージの構文チェック処理の記述を不要とし、システム標準処理としてチェックしてオンルーチンへ自動的に制御を移行する。これらにより、記述量を削減できるだけでなく、目的とする処理を中心にアプリケーションプログラムの論理を記述できるため、プログラムの読解性を向上できる。

4. 5 アプリケーション呼び出しとデータベースアクセスの位置透過機能

RPCにより、簡易言語アプリケーション間で位置透過な呼び出しができる。一般の3GLのRPC機能では、IDL(Interface Definition Language)による呼び出しインタフェースの定義と、IDLに基づく遠隔呼び出し用のスタブモジュールが必要であるが、この簡易言語間RPCでは、IDL定義やスタブモジュールを生成する必要がない。

また、リモートSQL機能により、簡易言語から位置透過なデータベースアクセスが可能であり、RPC機能と併用できる。

このRPC機能／リモートSQL機能は、VisualBASIC、PowerBuilder、SQLWindows、LotusNotes等の各種スクリプト言語で記述したクライアントプログラムからも利用できる。

これらの位置透過機能により、簡易言語でのシステム構築を前提とした上で、クライアント／サーバ型情報システムの各種の機能配置モデル(図1)を、業務に応じて使い分けることができる。

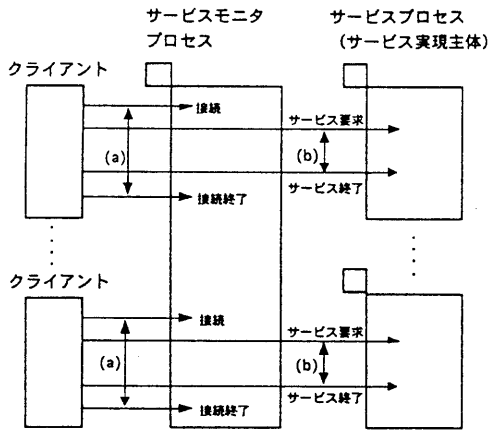
4. 6 会話型／リアルタイム型／バッチ型モニタ機能

非定型業務向けに比較的長時間のセッションを可能にする会話型制御、大量の定形処理のトランザクションを高速に処理するオンラインリアルタイム制御、日次・月次処理等の計画的一括型実行を可能にするバッチ制御、の仕組みを備える3種類のモニタ機能がある。これらの各モニタ上で同一の簡易言語が利用できる。このため、ほとんどの業務アプリケーションを簡易言語だけで構築できる。また、異なる実行形態での簡易言語の共用により、オンラインアプリケーション／バッチアプリケーション間で処理の共用／流用ができ重複投資が削減できる。更に、オンラインアプリケーション／バッチアプリケーションをコマンドラインから会話形態で起動して結果を確認できるため、段階的に簡易言語ルーチンの妥当性を検証でき、試験を容易化できる。

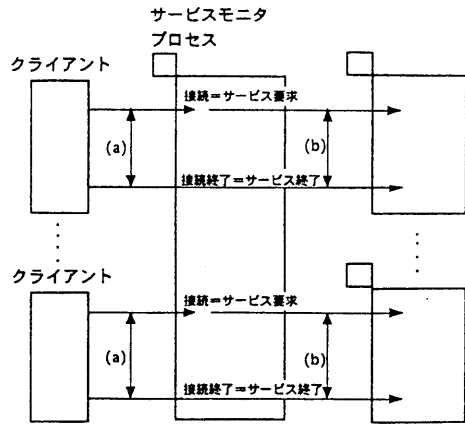
4. 7 トランザクションサーバ機能

VGUIDEのオンラインリアルタイム制御機構では、クライアントからの接続後、サービス要求とサービス開放の区間のみサービス主体プロセスをクライアントに占有させる方式(図3)を用いて、大量のクライアントを少ない資源消費で制御できる。

また、サービス主体プロセスはサーバ上で予め起動され、アプリケーションのメモリへのローディング、データベース等の資源確保を済ませた状態で待機する。このため、クライアントからのサービス要求時にはプロセスを割り当て



(a)と(b)は別々に制御可能(2階層の制御)
 クライアントの接続数とサービスプロセス数は無関係
 (a) 提案方式



接続の背景でサービス要求が行なわれる(1階層の制御)
 クライアントの接続数がサービスプロセス数となる
 (b) 従来方式

(注) (a) : クライアントとサーバの接続区間
 (b) : クライアントとサーバ間の会話処理可能区間
 → : クライアント~サーバ間プロトコル

図3 リアルタイム制御方式

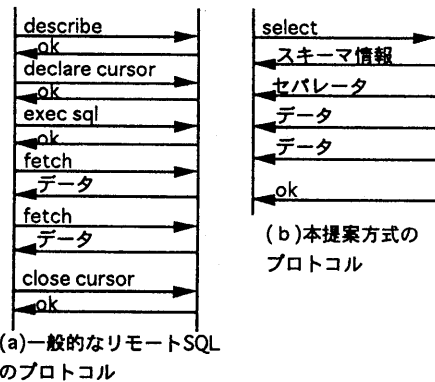
るとすぐに目的とする処理を行える。したがって、1つの要求をサーバで処理するための時間を削減し、大量の要求時にも高速な応答性能を保証できる。

4.8 リモートSQLの高速化機能

ネットワークを介したリモートSQLアクセスを高速に処理し、1つの要求に対するプロセス占有時間を削減するために、プレゼンテーション層以上の上位プロトコルとして検索結果のバルク転送を行うプロトコルを採用している(図4)。

4.9 マルチベンダ対応機能

VGUIDEの簡易言語アプリケーションはインタープリタ方式により実行されるため、プラットフォームとするOSに依存しない。このため、アプリケーションにまったく手を加えることなく、異なる機種、異なるOSのサーバへ搭載できる。また、DBMS間の差異もインタープリタが隠蔽しており、異なるDBMS環境への移行もアプリケーションを変更せずに可能としている。このため、



(a)一般的なリモートSQLのプロトコル

(b)本提案方式のプロトコル

図4 リモートSQLの高速化方式

ベンダに対する依存度の少ないシステムが構築できる。

クライアント機能は、Windows、OS/2、Macintosh、UNIX(Solaris)の各環境上でAPIが用意されているだけでなく、主要なGUI構築ツール(VisualBASIC、PowerBuilder、SQLWindows、LotusNotes)のスクリプトから利用が可能である。このため、表示層アプリケーションの構築環境を自由に選択できる。

5.1 アプリケーション記述量の削減

表2に同じアプリケーションに対する記述項目毎のCOBOLとVGUIDEの記述量の比較を示す。表2から、VGUIDEの場合、COBOLの約21.5%の記述量により同じアプリケーションを実現できることが分かる。特に、変数宣言、ブロック宣言、初期化処理、後処理、通信処理の削減効果大きいことが分かる。これは、簡易言語による入出力領域の隠蔽、定型処理の削減の効果である。また、カーソル操作の削減効果も大きい。これは、AP制御構造を単純化するためにフェッチ処理が不要であることの効果である。[7]

表2 AP記述量の記述項目毎の比較

項番	記述項目	説明	ステートメント数	
			COBOL	VGUIDE
1	変数宣言文	◇RDBMSのSCA域	1	0
		◇パラメータ設定域	3	1
		◇RDBMSのホスト変数	1	0
		◇検索結果件数一時域	4	0
		◇SQLエラーコード設定一時域	4	0
2	言語固有文	◇DIVISION, SECTION 等	13	0
3	構造化制御文	◇PERFORM, SECTION 等	14	0
4	SQL処理文	◇SQL関連変数の初期化	11	0
		◇RDBアクセスカーソル操作	28 ^{a)}	0
		検索	11 ^{a)}	4
		更新	11 ^{a)}	4
		COMMIT操作	8 ^{a)}	1
		◇DBアクセス終期化	8 ^{a)}	1
◇オンルーチン	0	4		
5	代入/ループ制御	◇代入/ループ制御	9	13
6	通信	◇検索結果/更新結果の通信	9	1
合計(ステートメント数)			135	29

(注) *1) RDBMSアクセスエラー処理も含む

5.2 リモートSQLの高速化

検索結果を一括転送するリモートSQLの高速化機能により、一レコードの転送毎に送信側と受信側の同期が取られる一般のリモートSQL方式と比較して、LAN環境で約1.5倍の応答性能が得られる(図5)。[7]

この方式はパケットの伝達に時間を要するWANを介した場合にも、大きな性能劣化のないことを特長としており、WAN経由での実用的なデータベースアクセスを可能にしている。

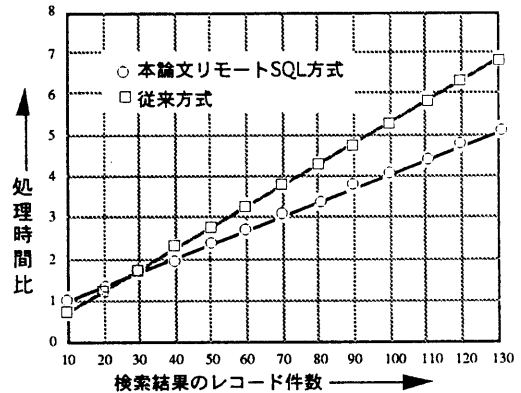


図5 リモートSQLでの検索件数と処理時間

5.3 トランザクション応答時間の高速化

以下の条件により、多端末からトランザクションを発生させ、従来型の会話制御とリアルタイム制御によるトランザクション応答時間(サーバ内でのトランザクション開始から終了までの所要時間)を比較した。この効果を、図6に示す。

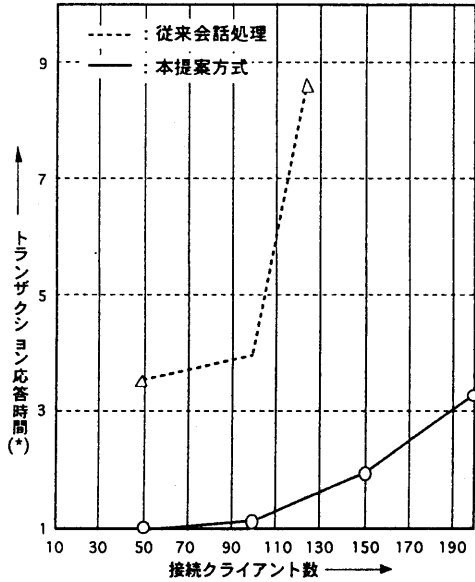
(条件)

- ・テーブル構成：200カラム×10バイト
- ・データ件数：1000件
- ・トランザクションの内容：

1レコードを検索後、更新

- ・各トランザクションからのアクセス対象レコードの競合無し

この結果から、リアルタイム制御でのトランザクション応答時間は会話型の1/3以下である。また、会話処理では端末数の増加により、1件毎のトランザクション応答時間が急速に増大し100端末以上では処理不能となるが、リアルタイム制御では200端末を超えてもトランザクション応答時間の増加は比例的であり、スケーラビリティが高いことがわかる。[5]



*) 本提案方式でのクライアント数50の場合の時間を1
 図6 クライアント数とトランザクション 応答時間の評価

5.4 業務応答時間の高速化

業務を模擬したオペレーションを多端末で実行し、業務セッションの開始から終了までの所要時間を、会話型制御とリアルタイム型制御で計測した。この結果を測定条件と合せて図7に示す。この結果は、会話型制御では100端末程度で業務セッション時間が急激に増大し、スループットが極端に低下するが、リアルタイム制御

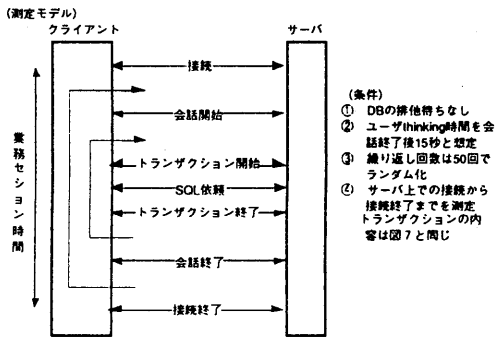
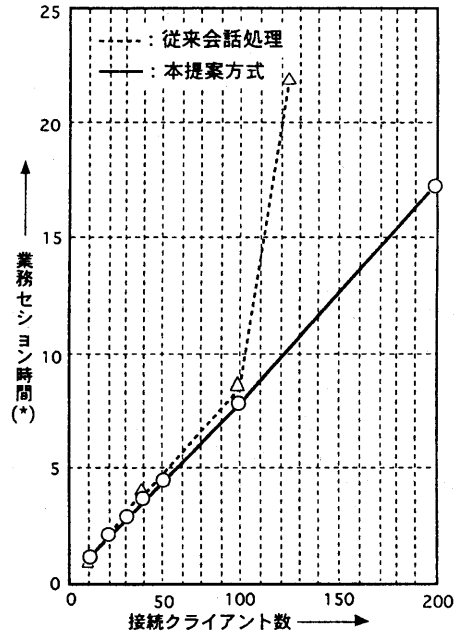


図7(a) 業務セッション時間の測定条件

では200端末を超えても一定のスループットを維持できていることがわかる。本方式は、後述する1000台以上のクライアントを接続する基幹業務システムの実運用において、応答時間2秒以内の性能を実証している。[5]



*) 本提案方式でのクライアント数10の場合の時間を1
 図7(b) クライアント数と業務セッション時間の評価

6. 適用事例

VGUIDEは、弊社の通信サービス販売・設備管理・故障受付窓口等の基幹系業務システム、経理・物品・施設管理等の社内システム、地域支店等での部門用システムに導入されており、社内外合わせて約30システム、380サーバ、8500クライアントの導入実績がある。

この中から適用事例として、通信サービスの申込み、工事、設備管理、故障管理、顧客管理等の業務をサポートする基幹業務システムを紹介する。図8にシステムの構成を示す。

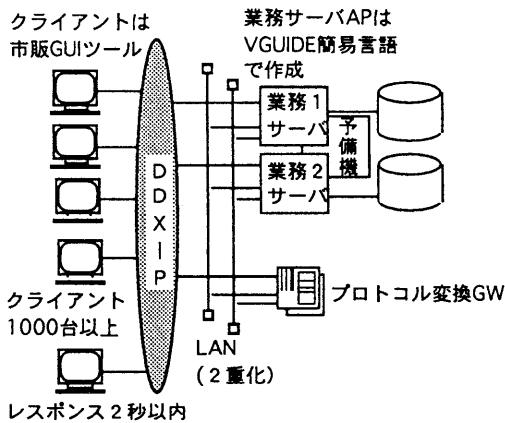


図8 適用システム例

当システムは、約100万のお客様回線を管理しており、業務別の2台の現用サーバと1台の予備サーバから構成される。1000台以上のクライアントが接続され、DDX-PのWANからゲートウェイを介してサーバに接続する。

サーバ上のアプリケーションは、VGUIDEの簡易言語で開発した。本システムでは、クライアントからRPCで簡易言語APを呼出し、簡易言語APからデータベースアクセスする。アプリケーション規模は、クライアントプログラム約200KLOC、サーバプログラム約100KLOC。開発期間は機能設計からサービス開始まで約10ヶ月であった。

本システムのレスポンス時間は、約200~300トランザクション/分の負荷に対して2秒以内である。

7. 考察

7.1 スケーラビリティ

NTT社内システムへのVGUIDEの適用事例毎のサーバ数とクライアント数の関係を図9に示す。現在、社内では約20システムに適用されている。これらの導入形態を分類すると以下の3種になる。

- (1) 高負荷型
- (2) 全国展開型

(3) 中小型

高負荷型のシステムは、1~数サーバで、数百から千数百のクライアントを収容する大規模トランザクション型のシステムである。現在、3システムの社内事例がある。この事実は、簡易言語による本格的な基幹業務システムの構築が可能であることを実証している。

全国展開型のシステムは、全国の支店(約200)に導入されているシステムである。現在、2システムの社内事例がある。これらのアプリケーションは、支店ごとに異なるマルチベンダ環境で稼働している。

中小型のシステムは、数十台のクライアントを収容する中小規模システムである。約15システムの社内事例がある。

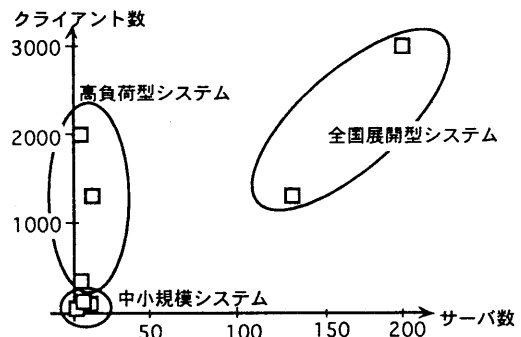


図9 VGUIDEのスケラビリティ

7.2 関連技術との比較

TPモニタや従来のDB操作簡易言語とVGUIDEを、生産性、性能、拡張性の観点から比較した結果を表3に示す。

TPモニタは性能面で非常に優れるが、C言語の関数ライブラリとして提供される機能を用いた高度なプログラミング技術が必要であることから生産性の面では劣っている。

DB操作簡易言語は、その簡易言語によって生産性が優れるが、リアルタイム処理向けの制御機能を持たないことから性能面で劣っている。また、遠隔DB型や遠隔表示型以外のクライアント/サーバモデルの形態では利用できない点で

拡張性が限定される。

VGUIDEは、簡易言語を利用して高い生産性が得られる。また、クライアント/サーバモデルのいずれの形態にも適用でき、マルチベンダ環境でのアプリケーションの移植性に優れており、柔軟性が高い。性能面ではオンラインリアルタイム制御機構により、高いスループットが得られる。VGUIDEは適用を通じ、簡易言語を利用した本格的な大規模トランザクションシステムを構築できることを実証している。

表3 関連技術との比較

	生産性	性能	拡張性
TPモニタ	△	◎	○
DB簡易言語	◎	△	△
VGUIDE	◎	○	◎

TPモニタと比較すると、VGUIDEの場合、複数サーバにまたがる分散トランザクションを扱えない、ファイルシステムとデータベースを同期させた救済ができないという問題がある。前者の問題については、多くの業務システムでは完全に同期して更新される必要のある分散データは少ない。従って、DBMSのレプリケーション機能やアプリケーション側で更新ステータスを意識することにより解決できる。また、後者の問題に関しては、救済すべきデータをすべてデータベース上に置くことによって解決できる。

8. まとめ

本稿では、データベース操作用簡易言語を備え、高負荷な基幹系業務にも適用可能なことを特徴とするクライアント/サーバ型情報システム環境VGUIDEについて、その技術的特長と適用状況を述べた。この中で、大量の要求に対して応答時間を保証するためのメカニズムを備え、トランザクションを含むデータベース操作が記述可能な簡易言語処理系によって、大規模な基幹系業務システムを低コストで構築できること

を実証した。

今後は、3層モデルに基づく設計法[9]に対するVGUIDEの適合性を検証して行く予定である。

謝辞

本検討を通してご指導いただきました長岡 満夫 通信ソフトウェア本部担当部長に深謝いたします。また、6章で挙げたシステム事例への適用に際して多大な協力をいただきました木村 裕 通信ソフトウェア本部主幹技師、ならびに日頃ご指導いただき、細谷 僚一 ソフトウェア研究所長および長野 宏宣 ソフトウェア技術研究部長に深謝いたします。

参考文献

- (1) 加藤・川手・長岡：UNIX上RDBアプリケーション制御方式の検討，情報処理学会第45回全国大会，3R-5，1992
- (2) 川手・黒川・長岡：垂直分散向き簡易言語の生産性向上要因について，電子情報通信学会秋季大会，D-70，1992
- (3) 元田・瀧野・長岡：HCI設計手法とその環境に関する一検討，情報処理学会第47回全国大会，3J-5，1993
- (4) 加藤・伊織・川手・長岡：UNIXトランザクション処理方式の評価，情報処理学会第47回全国大会，6C-2，1993
- (5) 加藤・元田・伊織・長岡：クライアント・サーバモデルと分散処理連携技術，NTT R&D，VOL.42，No.12，pp.1461-1472，1993
- (6) 長岡：エンドユーザコンピューティング環境，NTT R&D，VOL.42，No.12，pp.1451-1460，1993
- (7) 川手・黒川・倉持・長岡：分散型簡易言語処理技術，NTT R&D，VOL.42，No.12，pp.1473-1484，1993
- (8) 徳丸・元田・黒川：ビジネスオペレーション向け業務システム構築技術，NTT R&D，VOL.42 No.12，pp.1485-1494，1993
- (9) 高田・山本：3層クライアント/サーバシステム設計方法の提案，信学技報，KBSE95-6，1995