Vol.36 No.6 情報処理 June 1995

特別論説

情報処理最前線

# クライアントサーバ型基幹業務システム設計の評価方法†

岸田 一††

## 1. はじめに

オープンシステム，ダウンサイジング，インターネットワーキングなど技術の変化と進展によって，クライアント／サーバシステム（C/S システム）に代表されるような分散システムが普及しつつある．しかし，現在実用化されているほとんどの業務アプリケーションは従来のメインフレームを中心とした集中型の計算機上で構築されており，設計，開発に関するノウハウもこれらの環境を前提としたものである．昨今の技術変化の与えた影響はおおむねプラスの面が多いが，C/S システムの基本設計にとってはかなりマイナスに働いている．

## 2. C/S システムの基本設計

C/S システムの基本設計は，集中システムの設計技術にはない，リソースをどのように分散させるのかあるいは重複させるのかといったような複雑さを増大させる設計上の課題を持つ．ある処理には最適の分散パターンが別の処理には全く不適当なこともあり得る．システムインテグレーションの製品組合せパターンもかなりの数がある．一般に，定性的に妥当な C/S システム基本設計案はいくつか考えられるので，どの案を採用すればよいのかシステムの開発側も発注側も非常に迷うところである．

基本設計では，性能，信頼性，稼働率，コスト等を十分に考慮しなければいけない．ただし，これらの設計要素には“静的”なものと“動的”なものがある．信頼性，稼働率はシステム要素とシステム構成が決まると確率的に計算できるという意味で“静的”なものである．

しかし，処理性能は，基本システムアーキテクチャだけでなく，実際のシステムの使われ方やトランザクション特性とそれらの相互干渉に，大きく影響を受けるという意味で“動的”なものである．特に，システム全体を対象としたとき，机上で計算するには無理があり，性能予測は難しい．

このような複雑さと選択肢の多さ，“動的”な性質，および技術蓄積と経験の少なさのため，優秀な設計者でも不安を抱えながら C/S システムの基本設計をしているのが現状と思われる．逆に，自分自身の設計に自信と責任を持つために，業務アプリケーションシステムの骨組みをとらえた上で，システム基本設計上のポイントについて性能の定量的裏付けは必須である．“骨組み”というのは重要な点で，“部分”ではない．

ユーザサイドからすれば，自分の業務システムが C/S システムとして新規または再構築されるとき，どのような基本設計案があるのか，提示された設計案がどのようによいのか定量的，合理的な説明を必要とする．もちろん，他の事例に基づく経験や通信，RDB などに関する部分的な計測結果などでは，自分のシステム全体に対して提案された基本設計を合理的に判断することはできない．

C/S システムは常にシステム拡張を伴うものだが，拡張される部分がどうシステム全体の性能に影響するのか定量的に予測することなしに，システム拡張はできない．つまり，予測する技術なしではシステム拡張もできない．

性能評価に関連する技術にはベンチマークテストやシミュレーションがあるので，まず，それらを簡単に紹介する．次に，プロトタイピング手法

† An Evaluation Method for System Designs of Business Application Systems Based on Client-Server Architecture by Hajime KISHIDA (Information Technology Consortium).

†† （株）情報技術コンソーシアム研究開発部

に基づく性能評価方法について，ケーススタディをまじえ，紹介する．

## 3. 性能評価技術の現状

### 3.1 データベースベンチマークテスト

トランザクション処理性能評議会（TPC）により採択されたデータベースシステムのベンチマークテスト[1]として以下のものがある．

（1）TPC－ベンチマーク A[2]

1989年11月に公表されたベンチマーク仕様である．ダム端末，ワイドエリア・ネットワークを対象としていたDebit／Creditテストに由来し，端末としてインテリジェント・ワークステーションを利用したクライアント・サーバ・モデルにも対応，ネットワーク構成もホストとのローカルエリア相互接続を加え，tpc A-Wide，tpc A-Localといった仕様を準備した．

銀行窓口における顧客口座への入出金業務アプリケーションをモデルとしているため，単一の単純な更新主体のトランザクションになっている．使用される評価尺度は，応答時間の制約のもとで1秒当たりのトランザクション数（tps）で測られるスループットとtps当たりの価格である．

（2）TPC－ベンチマーク B[3]

1990年8月に公表されたベンチマーク仕様である．端末とネットワークを無視したバッチ型のTP1テストに由来している．DB処理に対する外部要因をできる限り除去した，単一の単純な更新主体のDBサービスを行うトランザクション処理からなる仕様で，元々DBベンダーが関心をもっていた．

使用される評価尺度は，DBサーバへの滞在時間の制約のもとで1秒当たりのトランザクション数（tps）で測られるスループットとtps当たりの価格である．

銀行の顧客口座への入出金業務アプリケーションをモデルとする．TPC－Aと違って，そのデータベース側の操作だけを模してバッチ・トランザクション・プロセスを使用する．

（3）TPC－ベンチマーク C[4]

1993年10月に公表されたベンチマーク仕様である．単純なOLTP環境を対象としていたTPC－Aに対し，より複雑なOLTP環境を対象にして様々なアプリケーションモデルへの対応を追加した仕様である．複雑，多様なトランザクションの同時実行や大きさ，属性，参照関係に広い変化をもったテーブルからなるDBに特徴を持つ．

評価尺度はTPC－Aと同様であるが，単位は秒（tps）ではなく分（tpm）を用いる．アプリケーションモデルは経営・販売・製品やサービスの配給といった多くの産業を想定している．

（4）TPC－ベンチマーク D[5]

1995年に公表される予定のベンチマーク仕様である．意志決定支援における参照処理を対象とし，複雑な問合せのトランザクションを使用し，長期に渡るDBアクセスも許す業務上の問合せ処理環境をシミュレートしている．

使用される評価尺度は，データサイズ当たりの問合せ実行能力（Qpp@Size）である．これは，ある該当データサイズにおける単一ユーザだけでなく，複数ユーザといった大規模化への潜在能力でもある．そしてQpp@Size当たりの価格である．

（5）TPC－ベンチマーク E（予定）

1995年に公表される予定のベンチマーク仕様である．以前の地域的なベンチマーク仕様に対し，地球規模の広域システムにおける処理を対象としている．

TPCからはクライアント・サーバ・モデルによるOLTPベンチマーク仕様も予定されている．TPC以外によるベンチマークテストとしては，以下のものが知られている[6]．

（6）ウィスコンシン・ベンチマーク

ウィスコンシン・ベンチマークはBitton, DeWitt, Turbyfillの1983年の論文（ウィスコンシン大学）により知られるようになったベンチマーク仕様である．

1981年に実装されたデータベース・マシンDIRECTの高速化の特性や「大学版」Ingresに対する相対的な評価に使うことを目的に開発された，関係データベース・システム用のベンチマークである．選択，射影，結合，集計のような基本的な関係演算における性能を測定する．

（7）AS3APベンチマーク

AS3AP（ANSI SQL Standard Scalable and Portable）ベンチマークはBitton, Orji, Turbyfillの論文（イリノイ大学）により知られるようになった．単純な環境における関係問合せのベンチマー

クとして標準となったウィスコンシン・ベンチマークに対し，マルチユーザやユーティリティ併用等，混合負荷環境における統合的な性能測定を目的としたベンチマークである．

評価尺度としては，いかに大きなデータ規模まで規定の処理要求を満足できるかといった「等価データベース・サイズ（equivalent database size）」，複数システム間の比「等価データベース比」，DBMSの全価格を等価データベース・サイズで割った「1メガバイト当たりのコスト」がある．

### 3.2 シミュレーション

システムの性能評価のための技術としてシミュレーションがある．一般的なシミュレーションプロセスは以下の手順で進み，各種シミュレーションツールはこれらのプロセスを支援する機能を提供している[7]．

1. システムの記述
2. システムの抽象化とモデルの記述
3. データ収集
4. 解析方法の選択
5. シミュレーションプログラムの開発
6. プログラムのデバッグ
7. プログラム対モデルの確認
8. モデル対システムの検証
9. シミュレーション実行と解析
10. 問題の解析

（1）モデルの記述とプログラム（プロセス2,4,5）

モデルの記述，その詳細化およびシミュレーションプログラムの開発は，ツールにもよるが，主にシステム構成要素とシミュレーションパラメータを定義することで行われる．これらの定義は，GUIベースの対話的な入力や選択などで済むように簡易化されているが，ツールで提供されているシミュレーション言語も使用される．

（2）デバック・エイド（プロセス6,7）

シミュレーション・プログラムに対してエラーの報告，トレース，ダンプ等の支援を行う．

（3）シミュレーションの実行，計測（プロセス9）

モデル記述されたパラメータによりファシリティーのコントロール，イベントのスケジューリング，確率変数に基づくタスク・イベントの発生等の制御を行い，自動実行，計測を行う．

（4）レポート，自動解析（プロセス9,10）

シミュレーション実行時の性能計測に基づくレポート出力，ツールによっては問題点の自動解析やアニメーション出力による問題点表示といった支援を行う．

### 3.3 既存技術の課題

データベースのベンチマークテストはデータベースシステムの性能評価ではあるが，必ずしもC/S型業務システムの性能予測にならない．基準を定めて性能比較することは重要で，有益なことであるが，同時にベンダ間の性能競争になりがちである．性能を出すために違反にならない様々な“トリック”をするので，いっそう実システムから遊離してくる．システムインテグレーションの第1次フィルタには利用できると思われる．

シミュレーションによるシステムの性能評価，予測には，モデルの記述の詳細化の問題（プロセス2），データ収集の問題（プロセス3）がある．シミュレーション精度を上げるためには，モデルを詳細化する必要があるが，モデル設計者の高いスキルが要求される．また，実システムの能力や負荷のデータを収集，推定するのも容易ではないし，不正確になりがちである．モデル対システムの検証（プロセス8）も，実システムが存在しない場合，推測に頼らざるを得ない．

## 4. 基本設計性能評価ケーススタディ

プロトタイピング手法に基づく基本設計性能評価方法を使った事例を紹介する．このケーススタディは，（株）情報技術コンソーシアムで実施されている「広域分散ソフトウェア生産技術開発プロジェクト」の成果の一部を簡単にまとめたものである．このケーススタディのために構築したテストベッドシステムは，INS64で接続された3台のUNIXサーバからなり，それぞれにRDBとTPモニタが実装されている．

### 4.1 販売・在庫管理システムのモデル

このケーススタディは，ある販売・在庫管理システムをモデル化したものであり，以下にその処理とデータ構造（**図-1**）の概要を示す．

**受注処理**：クライアントからの注文を受け，ローカルサイトの部分在庫テーブルの在庫量を照会し，在庫が引当可能であれば更新，在庫不足であれば引当待ち品発注テーブルへ発注データをイン

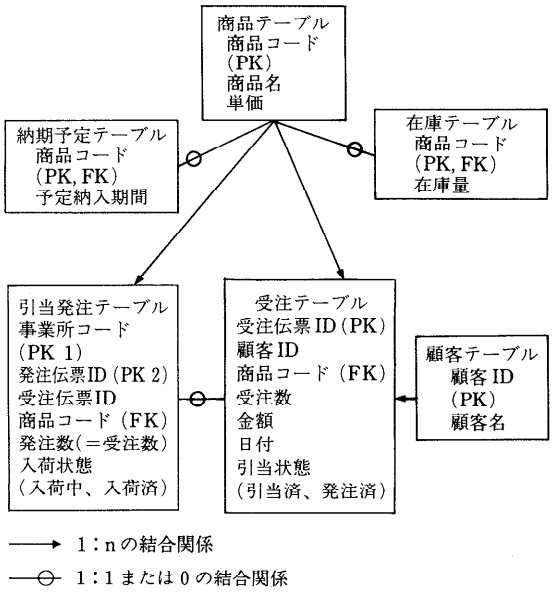


**図-1 データ構造**

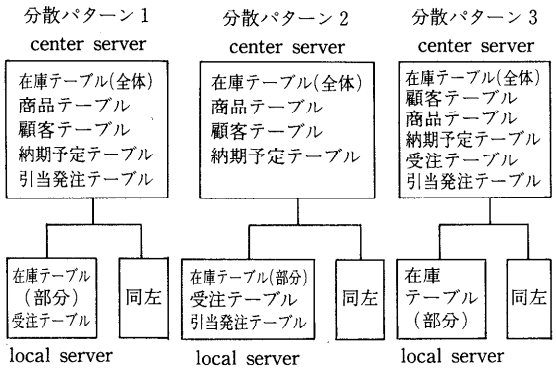


**図-2 分散化パターン**

サートし，いずれの場合も受注テーブルへ引当情報をインサートする．

**在庫照会**：部分在庫テーブルを照会し，在庫量を返す．

**納期照会**：商品コードをキーとして，部分在庫テーブルと納期予定テーブルを参照し，納期日を返す（結合表検索）．

**引当状態照会**：受注テーブルと引当発注テーブルの結合表検索を行い，受注テーブルの引当状態と，引当発注テーブルの入荷状態等を出力する処理である．

**顧客情報照会**：顧客コードをキーとして，特定顧客の受注，引当待ち品発注，商品（価格）情報を出力する．

### 4.2 分散化のパターン

センタ（本社）と複数ローカルサイト（支社，倉庫）からなる販売・在庫管理システムで，前述の図-1の各テーブルを分散配置する．ここでは，**図-2**の三つの分散化パターンを設定した．分散パターン1は受注管理と在庫管理の自治性は高いが，在庫不足時の商品発注はセンタ側に任せている．分散パターン2はローカルサイト側で引当商品の発注も行い，分散パターン3では在庫管理だけを行っている．なお，全パターンとも在庫保有ヒット率は70％とする．

分散パターン1では，ローカリティの高い受注処理は速いが，分散ジョイン検索となる引当状態照会は遅いと予想される．分散パターン2では，受注テーブル，引当発注テーブルともローカルサイトにあるため受注処理はかなり速いと思われる．分散パターン3では，テーブルが集中配置されているため，引当状態照会のように広範囲にテーブルをスキャンするバッチ的な処理に強いと思われる．

異なる処理を同時に走らせる事は，各処理の応答時間に互いに影響するし，実際の運用に近い．このケーススタディでは，受注処理：16，納期照会：8，在庫照会：8，引当状態照会：1，顧客情報照会：1，の割合で処理を混在発生させている．

### 4.3 評価結果

分散パターン1，2，3の各処理の性能計測結果をまとめたものを**表-1**に示す．在庫テーブルはセンタサーバとローカルサーバに部分重複して分散されているので，その同期更新をする場合としない場合に分けて測定した．数値は在庫照会（実測値でパターン間の差無し）を1としたときの相対値である．

受注処理は，在庫テーブル，受注テーブル，引当発注テーブルをアクセスする．在庫テーブルの同期更新なしの場合，ローカリティの観点から，分散パターン2が最もよい性能を出し，分散パターン1が続いている．分散パターン3は，WANを経由してセンタサーバにレコードインサートするため処理に時間がかかり，大きく性能を落としている．

同期更新を行う場合，ローカリティが低くなり，

表-1 各分散パターンの比較

| | 同期更新なし | | | 同期更新あり | | |
|---|---|---|---|---|---|---|
| 分散パターン | 1 | 2 | 3 | 1 | 2 | 3 |
| 受注処理 | 6.7 | 3.2 | 22.2 | 10.5 | 11.2 | 24.1 |
| 納期照会 | 4.2 | 2.7 | 7.8 | 5.8 | 6.7 | 8.0 |
| 在庫照会 | 1.0 | 1.0 | 1.0 | 1.1 | 1.2 | 1.3 |
| 引当状態照会 | 21.4 | 5.9 | 16.1 | 24.1 | 11.0 | 24.1 |
| 顧客情報照会 | 27.8 | 8.5 | 9.7 | 30.2 | 15.1 | 14.5 |

その分，分散パターン2のメリットがなくなるが，受注処理が分散パターン1と同程度になるとは予想外である．また，納期照会は直接関係してないが，同期更新によるセンタサーバの負荷増大の影響を受けていると思われる．ローカリティを高くし，処理効率を高くする意図で分散パターン2を設定したが，以外と簡単にそのメリットがなくなっている．

引当状態照会，顧客情報照会は受注テーブル，引当テーブル，顧客テーブルのジョイン検索である．分散パターン1では，分散ジョイン検索となり，相応に処理時間がかかっている．分散パターン3は，必要なテーブルは同一サーバ上に配置されて処理効率は高そうであるが，逆に，すべての処理を実行するため，過負荷状態および排他制御の影響で処理時間がかかっているものと推測される．

## 5. システム基本設計の評価手順

プロトタイピング手法によるテストベッドシステム上でのC/S型基幹業務システム基本設計の定量的評価は，以下の手順で行われる．

(1) 評価用システム論理モデルの作成

一般的に，情報システム設計方法論[8]，ERモデル[9]，構造化分析などの手法による業務分析の結果，業務システムモデルとそのデータ構造が作成される．これらを基にして，評価用システム論理モデルは，C/S型システム基本設計の定量的評価のために，主要なトランザクション処理，その特性，バッチ処理，データ構造に関してさらに抽象化したものである．

ボトムアップ的に主要なものだけを先に取り出して評価用システム論理モデルを作成することも可能である．目的は，定量的評価のためのモデルを作成することなので，そもそも，業務システム機能上あまり重要でない機能に対応する処理プロセスあるいは使用頻度の低いデータは取り除かれる．業務システムの基本的かつ重要な部分の構造を“荒い”が“速く”把握できるまたは結果として分析できていることが重要である．簡易版インフォメーションエンジニアリング的手法でよいと思われる．

(2) 評価用システム論理分散化モデルの作成

このモデルは，評価用システム論理モデルで抽出したデータと処理プロセスの論理的分散パターンを定義したものである．“論理的”とは“分散可能”であるという意味で，物理的実装としてはシングルサーバで実現されることもあり得る．

評価用システム論理分散化モデルは，業務組織構造的な観点とシステム技術的観点から複数モデル作成される．ロケーション－ビジネスプロセス－データの親和性検証による分散配置だけでなく，分散化による性能向上，信頼性向上などのために意図的なデータ重複を含む分散もありうる．

分散化の観点を以下に示す．負荷分散による性能向上やアベイラビィティの向上などの分散化メリットがある一方，分散処理オーバヘッドも発生する．一般的に，単純で負荷の軽いシステムを分散化すると性能的にはデメリットの方が大きい．

(a) 独立な処理とデータ

ERモデルとして独立している，またはある処理群とあるデータ群が密接に関係し他との独立性が高い場合，それらをセットであるノードへ配置する．負荷分散が行われ，各処理はCPU，DBMSなどの資源をより多く占有することができる．完璧に分離できる場合は単純であるが，ノードをまたがるような処理が部分的にでも発生する場合，参照整合性処理，重複データ更新処理などのオーバーヘッドが生じる．

(b) データの水平分割

地域関連レコードなどのようにデータアクセス局所性がある場合，テーブルをレコード単位で水平分割し，地域ごとに分散配置する方法である．ローカリティが高い場合処理効率は良くなるが，低い場合はテーブル全体または他地域データに対する処理が発生し効率を下げる．また，ローカリティを高くするために，部分的にデータを重複さ

せる場合，データの一貫性を保つために同期更新処理が発生する．

(c) データの垂直分割

特定のフィールド群にデータ・アクセスの局所性がある場合に，テーブルをフィールド単位で垂直分割する方法である．たとえば，顧客情報テーブルに対して，マーケティング部門とアカウンティング部門では必要とする顧客データ属性が異なるので，垂直分割可能となる．元々，正規化されたRDBのテーブルからなるデータ構造を持つシステムはこの形態である．分割後は，ジョインオペレーションやインテグリティ制約に関する処理に対して分散処理オーバーヘッドが発生する．

(3) 評価用システム物理モデルの作成

前述の二つのモデルが論理設計であるのに対し，評価用システム物理モデルはその物理設計に対応する．このモデルは，LAN/WANネットワークシステム，UNIX機やNT機などのハードウェア，RDBやOLTPなどのミドルソフトウェアのインテグレーションを前提としてシステム構成を定義する．

インテグレーションされる個々の製品の選定には，ベンチマークテストや性能評価レポートが参考になる．稼働中のシステムを拡張する場合も多く，すでに設置されている機器に依存してシステム物理モデルを作成することもある．論理モデルと同じく，この物理モデルも性能評価用モデルである．実際のシステムと同じであることが望ましいが，必ずしも一致するわけではない．

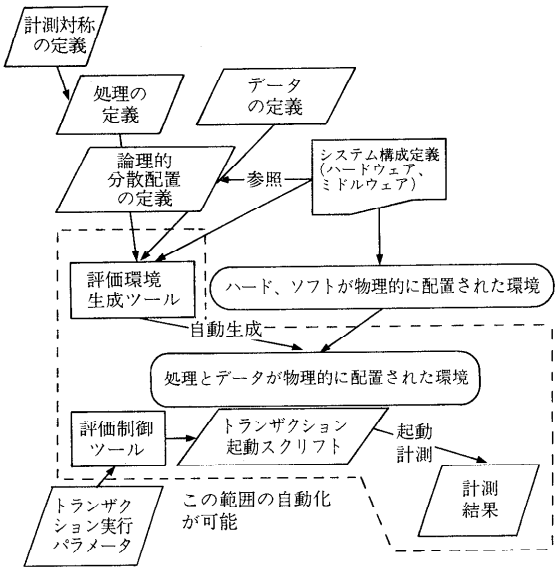


**図-3** 評価支援ツールの概念モデル

最後に，システム論理分散化モデルで得られたデータとプロセスの分散配置をこのシステム物理モデル上にマッピングし，評価用のシステム基本設計が完了する．

(4) システム基本設計の性能評価

システム物理モデルのシステム構成定義から基本設計評価用テストベッドシステムを構築する．この環境上で，**図-3**のような評価支援システムによりトランザクションを発生させ，その性能を測定する．測定実行前には，インデックスのアサイン，データのクラスタリングやデータ読み込みやログ書き出しのバッファサイズ，データキャッシングサイズなどの各種パラメータチューニングやモードの設定も行う．

評価支援システムは，評価環境生成ツール，評価制御ツールと各種定義情報からなる．評価環境生成ツールは，データの定義からテーブルを作成し，論理的分散配置の定義から処理とテーブルの分散配置を行う．評価制御ツールは，トランザクション実行パラメータの定義から，プロセスの起動，トランザクションの発生，処理時間の測定，プロセスの終了などを行う．

前述のケーススタディでは，システム論理モデルで定義されたトランザクション処理プログラムはC言語と埋め込みSQL文で記述されている．これは，2フェーズコミット，ディレードアップデートなどの分散データ制御メカニズムの選択や，ストアードプロシージャ，TPモニタ，標準・拡張SQLによる通信などのサーバ／クライアント通信メカニズムの選択によって，トランザクション処理プログラムのコーディングが異なるためである．これらのメカニズムをパラメータによって選択，指定できるようにすることは，今後の課題である．

## 6. おわりに

クライアント／サーバ型基幹業務システムの基本設計案を性能に関して定量的に評価する必要性といくつかの性能評価方法について概説した．紹介したケーススタディは基本的にプロトタイピング手法によるものである．実際にプロトタイピン

グを実行することによって各トランザクション処理の速度の程度を把握でき，これから構築または拡張するシステムの処理性能を予測できることは，大きなメリットである．一方，その手順，内容と有効性についてはまだ試行錯誤の段階で，モデルの作成から性能測定にかかる時間も重要な課題である．システム基本設計評価技術の今後の発展に期待する．

## 参考文献


1) Shanley,K.:TPC Background,Transaction Processing Performance Council (TPC) (May 1994).
2) TPC Benchmark A Standard Specification, TPC (June 1994).
3) TPC Benchmark B Standard Specification, TPC (June 1994).
4) TPC Benchmark C Standard Specification, TPC (Oct.1993).
5) Raab,F.: TPC Benchmark D Working Draft 7.0, TPC (May 1994).
6) Gray,J.編, 喜連川優,渡辺榮一監訳：データベース・ベンチマーキング, 日経BP社 (Dec.1992).
7) M.H.マクドゥガル：シミュレーションによるコンピュータ・システムの性能評価, 工学社 (Mar.1990).
8) Martin,J.:Strategic Data-Planning Methodologies, Prentice-Hall (1982).
9) Chen,P.:The Entity-Relationship Model: Toward a Unified View of Data,ACM TOD,Vol.1 (1976).


(平成7年4月7日受付)


岸田 一 (正会員)

1955年生．1980年広島大学大学院環境科学研究科修士課程修了．同年沖電気工業（株）入社．1984年カーネギーメロン大学訪問研究員．データベース管理システム，分散オペレーティングシステムの研究開発，金融ユーザ向け各種システムの開発などに従事．現在，（株）情報技術コンソーシアム研究リーダ．ACM会員．