4N－05　分散システムの性能設計法に関する研究

米田　誠次　佐藤　洋一　蔵元　英雄

(株)情報技術コンソーシアム　研究開発部

1．はじめに

納入時にユーザの要求性能を満たすことができず，性能が問題となる分散システムが増えている．分散システムで性能問題が多発する原因はいろいろ考えられるが，以下のような事例から，性能に対する見識不足や楽観など，主として性能問題に対する取り組みの遅れに起因していると考えられる．

- ●分析の最終段階から設計の初期段階にかけてユーザの要求性能が明示されないケースが多く，このようなとき性能の検討は先送りされる．
- ●性能は実装設計が完了するまでは不明であるとして，それまで検討されないことが多い．
- ●実データ程度の規模とユーザ数でのテストは運用開始直前になって受け入れテストではじめて実施されることが多く，この時期に性能問題が顕在化しても解決することはほとんど不可能である．

本稿は分散システムの性能問題に対して，実現可能性の検討，性能プロトタイプ法および性能シミュレーション法と名づけた３つの性能予測法を通しての対処法（性能設計法）を提案するものである．

2．性能設計法の概要

以下に筆者らの提案する性能設計法の概要を示す．

（１）分析の最終段階から設計の初期段階でユーザの要求性能を定量的な目標値として設定する．

（２）ある実装環境[1]を仮定したとき，ユーザの要求性能の実現可能性を判断する手段を準備する．

（３）ユーザの要求性能を実現するための実装設計を必要に応じて性能プロトタイプ法および性能シミュレーション法を織り交ぜながら進める．

実現可能性検討，性能プロトタイプ法および性能シミュレーション法はともに実現性能を予測しながら設計を進めるものであり，開発の早期段階から性能問題に対する取り組みを可能とする．

3．性能基礎データ収集と実現可能性の検討

分散システムの性能を詳細に検討する前に，要求性能に対する実現可能性を大まかに検討する手段が欲しくなる．この検討は実現可能性の低い環境で詳細に検討するなどの無駄な作業を排除すると同時に，コストと要求レベルのバランスをユーザに確認する手段ともなる．その手順を以下に示す．

（１）分析段階で，主要エンティティとそのCRUD操作[2]の系列としてのシステム機能とその負荷が確定しているとする．

（２）CRUD操作に対する性能は実装環境が決まれば，システム機能の実装詳細が決まらなくても計測できる．これを性能基礎データと呼び，これを収集する．

（３）システム機能の中で高負荷のものを上位からいくつか抽出する．

（４）性能基礎データの収集結果を（３）の高負荷機能に当てはめて，これらの機能がユーザの要求性能を満たすかどうかを判断する．

（５）ユーザが性能問題を十分に認識できず，要求性能を設定できないときは，いくつかの実装環境に対する高負荷機能の性能を実現可能値として提示し，どれかを性能目標値として設定してもらうことも考えられる．

4．性能プロトタイプ法

性能プロトタイプ法は特定の実装環境，実装方式で高負荷のシステム機能を実現してみて，それがユーザの要求性能を実現できているかどうかを予測するもので，対象機能に関するベンチマークテストによる性能予測法というべきものである．これを実行するには以下のようなものが必要となる．

（１）対象機能を実行するための（擬似）データ

（２）時間計測ツール

A Study on Performance Design Method for Distributed Systems by Seiji Yoneda, Yoichi Satoh and Hideo Kuramoto
Information Technology Consortium Corp. (ITC) Kiba Koen Bldg., 5-11-13 Kiba, Koto-ku Tokyo, 135-0042 JAPAN

（3）資源消費状況モニタリングツール

性能プロトタイプ法ではサーバマシンやクライアントマシンの台数，ユーザ数が実システムと比べて少ない環境（縮退環境）しか準備できないことが多く，その性能予測の精度には限界がある．

5．性能シミュレーション法

性能シミュレーション法は実運用環境でのシステムの全体的な特性やシステム条件を変えたときの応答性の変化をコンピュータシミュレーションで予測する目的で実施するもので，特に大規模システムで性能プロトタイプ法の環境と実運用システムの環境との乖離が著しいと判断されるときに適用する．

ここではシミュレーションモデルを，アプリケーションモデル，ミドルウェアモデルおよびネットワークモデルの３階層の構成で検討した．

（1）アプリケーションモデル：アプリケーションとしての論理機能を抽象化したもの．アプリケーションが扱うデータとその操作を規定する．

（2）ミドルウェアモデル：データベースサーバ，TP モニタ，アプリケーションサーバおよびクライアントの機能をモデル化し，トランザクションの流れを規定する．複数クライアントからの同時実行性は待ち行列でモデル化する．

（3）ネットワークモデル：ミドルウェアモデルで規定した機能の配置およびその間のネットワークを規定する．

性能に関するデータはミドルウェアモデルへの組み込み機能として，性能基礎データを入力する．

6．例題プロトタイプへの適用結果

例題として在庫問題を取り上げ，高負荷の機能として，在庫受注（引当て）処理を想定した．この機能のプロトタイプを WindowsNT，Tuxedo，Oracle 上で構築し，Pro*C による埋め込み SQL で CRUD 操作を Tuxedo から直接呼び出した実装（CRUD 版）と，引当て処理を Oracle のストアドプロシージャで実現してそれを Tuxedo から呼び出した実装（SP 版）で応答時間を計測した．またこれらの計測に対応する性能シミュレーションを実施した．プロトタイプの計測結果とシミュレーション結果の一部を図１に示す．図から SP 版はあまり適合性がよくないが，CRUD 版にはかなり良好な適合性が認められる．

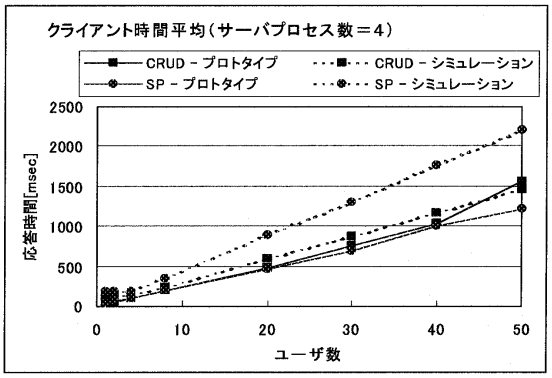

図１　計測の結果とシミュレーションの結果

7．結論

分散システムの性能問題に対処する方法として，

（1）性能基礎データの収集と実現可能性の検討

（2）性能プロトタイプ法

（3）性能シミュレーション法

の３つを検討し，これらが性能設計の中の性能予測法として有効であるという見通しを得た．これらの方法を機能させる前提として，分析の最終段階から設計の初期段階でシステム機能の確定とともに要求性能と負荷を定量化し，実現可能な数値目標を設定しておく必要がある．ソフトウェアの品質を構成する要素として，機能，性能，ユーザインタフェースおよび保守性などがあげられる．性能はソフトウェアの品質を構成する重要な１要素でありながら，設計法としてこれまで十分には検討されてこなかった．ここに述べた設計法は，分散システムの性能問題に対してかなり有効であると思われる．

謝辞

本稿は（株）情報技術コンソーシアムが情報処理振興事業協会より委託を受けて実施した研究成果[3]の要約である．本研究の実施に際して京都大学，垂水浩幸助教授のご指導および（株）構造計画研究所，（株）沖ソフトウェア，（株）日立情報システムズ各社のご協力を得たことを記して謝意を表する．

---

[1] ここではハードウェア，OS，ミドルウェア（DBMS,TP モニタなど）の実現可能な１つの組み合わせを実装環境とよぶ．

[2] CRUD は Create，Read，Update，Delete の頭文字をとったもの．SQL の Insert，Select，Update，Delete に相当する．

[3] 「分散システムの体系的な設計技法に関する研究開発」，平成 12 年 2 月報告．