

# 情報処理システムの性能評価(3)

紀 一誠

## 1. はじめに

前回までに情報処理システムの性能評価技術の概要について紹介した。本稿では、これらの技術を用いて行われる実際の性能評価作業の進め方について述べる。システムの性能を的確に把握する事は、システムの設計・開発を発注する企業と、これを受注し作業を行うメカ企業の両者にとって極めて大切であるが、システムの性能に最終的に責任をもち、顧客が満足するシステム性能を実現することは設計・開発を行う企業側の責任である。このため、システム性能評価作業の進め方に関するノウハウや必要とされる技術は主としてシステムの設計・開発を行うメカ企業において発展し蓄積されてきた。本稿で紹介する技法や考え方はこのような背景をもっている。

## 2. 性能評価作業の進め方

システム性能評価作業はともすればシステム開発作業の陰に隠れ、十分な対応がなされないままに推移し、開発がかなり進んだ段階で問題が発生しその対策に多大な工数をとられてしまう傾向がある。特に、大規模なシステム開発においては、このような後向きの工数の発生は膨大な損失と納期遅延を引き起こす原因となり、また性能改善対策のため予期せぬ多大な費用が発生することにもなる。大規模なシステム開発においては性能改善作業無しにサービスインできる事はない。そのため、開発プロジェクトの中で性能評価作業を当初から適切に位置付け、組織的に作業を実施して行く事が必要である。

システム性能評価作業は対象となるシステムの規模や開発の段階によっても異なるが、およそ以下のようなステップに要約できる。

- 利用者特性の把握：負荷(トランザクション、ジョブ)の種別、発生率、運用形態、最繁時負荷、他。
- ワークロードプロファイルの設定：各負荷毎のシステム資源使用量(例：1トランザクション当たりのCPU時間、DBアクセス回数、他)。
- システム性能評価指標の算定：トラヒック計算、資源使用率、応答時間、スループット、等の算定。
- 分析・評価：問題点の抽出、対策案の検討、他。
- 性能改善作業とその効果の把握：改善案の実施、効果の測定または推定。
- 性能試験：システム性能の最終確認。

これらの作業項目のいくつかはシステムの提案から設計・開発を経て運用に至るまでの間、何回も繰り返し行われ、またその段階に応じて目的や精度に対する要求、利用できるツール等も異なっている。性能評価作業の中で最も困難で労力を要するのは、ワークロードプロファイルを正確に把握するために必要とされる作業である。この事は、例えば開発プログラムの本数約1万本、プログラムステップ総数が数メガライン、開発工数も数千日月といったシステムの開発において、基本設計段階から性能試験の段階にいたる各開発段階で必要なワークロードプロファイルを組織的に収集することの大変さを考えれば想像できよう。さらには、オープン化の進展に伴い、開発会社が異なる様々な製品を利用しながら構築されるシステムも増加しているが、一般にこれらの製品に関する性能データは殆んど公表はされないため、ワークロードプロファイルを知るためには大きな困難が伴う。

きの いっせい NEC C&C 研究所  
〒 216 川崎市宮前区宮崎 4-1-1

### 3. シングルプロファイル法

前章に述べたシステム性能評価作業を効果的に実施するためには、まず性能評価の方法論が明確に示される事、ならびにその方法論に沿って実行される性能評価作業を効率的に支援するためのツール群が完備している事が必要である。本章ではこのような背景のもとに提案された「シングルプロファイル法」[1, 2, 4, 5]とその考え方に沿って開発された性能評価ツールの紹介ならびにシングルプロファイル法の検証例を紹介する。

#### 3.1 シングルプロファイル法の考え方

前章に示したように、的確な性能評価作業を行うためには次の2点への配慮が本質的に重要である。

- 性能評価の基本となるワークロードプロファイルをいかに正確に把握するか。
- 得られたワークロードプロファイルからいかに的確にかつ簡単にシステム性能を予測するか。

これらの要請に対して、シングルプロファイル法では次のような考え方を提案している。まず、前者のワークロードプロファイルの把握については、シングルプロファイル可能な限り実測する。後者のシステム性能の予測はこのデータに基づき待ち行列網モデルを利用して行う。ここで、シングルプロファイルとは、システムに負荷としてかかる様々なトランザクションについて、それぞれのトランザクション当たりを使用する種々のシステム資源の使用時間データの事を表している。また、システム資源とは次の2種類の資源の事を意味している。

- 競合資源：不特定多数の利用者から利用され競合状態を起こす可能性のある資源。典型例としてCPU。
- 遅延資源：利用者が占有して使用することができ、他の利用者との競合を発生しない資源。性能的には単に使用時間分の遅延時間を発生するだけである。

シングルプロファイル法においては、システムに負荷としてかかるトランザクションを必要に応じていくつかの典型例に分類し、それらのトランザクション毎にシングルプロファイルを調べる。即ち、トランザクション毎に、その処理を開始してから終了するまでの間に使用するシステム資源の種類とその使用時間を調査する。シングルプロファイルを収集するためには原則的には測定技術を用いる。シングルプロファイルを得るために行われ

る測定では、システム資源に競合が発生しない状態、即ち1クライアント(端末)のみがサーバに接続された状態でトランザクション処理を行う環境のもとに、そのトランザクションが使用するシステム資源の使用時間の測定を行う。このため、シングルプロファイルの測定環境をは大変に簡単であり、多数のクライアントを接続してトランザクションを発生させる大掛かりな環境を作る必要は無い。一方、簡素な構成であるとはいえ実測を行うので、その測定環境を整える必要がある。この測定環境を構築するためにはプロトタイピング技術を用いる。このように、シングルプロファイル法においては性能評価の基礎になるシングルプロファイルデータをプロトタイピング技術と測定技術を用いて収集する事を基本とする。その理由は、第一にオープンシステム的环境下ではマルチベンダによる製品構成をとる例が多く、性能データは実測してみる以外に信頼のおける調査の方法が見つからない事、第二にシステムの低価格化が進んだため、実機環境を整えプロトタイプモデルを構築することが比較的容易にできるようになって来ていることである。また、性能測定を目的とする環境であるため、システム動作のみが実システムの特徴を反映していればよく、そのためのプロトタイプモデルの作成は一般には容易であることもこの方法の利点の一つである。

一方、シングルプロファイル法においては、複数のクライアントから非同期にトランザクション負荷が発生してくる状況における応答時間や資源利用率等の性能評価指標の予測を行う部分は、待ち行列理論の成果を利用した解析的な方法を採用し、実機環境を用いた測定実験を行う事を避けている。その理由は次のようなものである。第一に、複数クライアントが接続された状況での測定を行う事はエミュレータの利用を伴う比較的大規模な測定環境を整えねばならず、準備と測定実験にかかる工数と費用が大きい事、第二に、性能予測作業は、設計・開発の途上で発生する様々な設計変更、性能改善条件に対応して、短期間・少工数で迅速に繰り返し行いたい場合が多く、この要請に応えるためには待ち行列理論による解析的な性能予測が最も効率的である事による。

以上に紹介したように、シングルプロファイル法は測定技術と予測技術の各々の長所を組み合わせる事により、簡単にしかも精度の高い性能評価作業が行える事を目的としている。このシングルプロファイル測定法の考え方の基本を要約すると次のように表現する事ができる。

# 1 を聞いて n を知る。

ここで、1 を聞くとはシングルプロファイルプロトタイプ技法を用いてシステム資源の競合が無い状態での実測により得る事を意味している。また、n を知るとは、測定によって得られた質の良い基礎データ(シングルプロファイル)に基づき、接続されている多数のクライアント(n clients)から非同期にトランザクション負荷がかけられてきた場合の状況を待ち行列網モデルを用いて予測を行うという事を意味している。シングルプロファイル法による簡単な性能評価の例を次に示す。

例：図1に示されるように、データベースサーバシステムに n 台のクライアントが接続されているシステムの性能評価を考える。サーバはデータベース(DB) DISK 1台とトランザクション処理内容の履歴を記録するログファイル(LOG)用のDISK 1台の計2台のDISK、およびプロセッサ(CPU) 1台から構成される。各クライアントから発生するトランザクションは、サーバ内でアプリケーションプロセス(AP)と二つのデータベースシステムのプロセス(DB Writer, Log Writer)により処理が行われる。始めにこのトランザクションのシングルプロファイルを測定する。その結果は表1に示されるように、全体でCPU使用時間が1トランザクション処理あたり平均30ms、DB DISKへの物理アクセス回数は2回の読み出し、1回の書き込み、LOG DISKへは1回の書き込みがあった。また、DB DISKへのアクセス時間は1回平均30ms、LOG DISKへのアクセス時間は順次書き込みのためシーク時間が短く平均15msであった。このときのサーバ側の1トランザクション処理時間の内訳(シングルプロファイル)を図2に示す。こうして得られたシングルプロファイルデータに基づき、図3に示される待ち行列網モデルを用いて各種の性能指標を算出する事ができる。一例として、クライアント数の増加に伴い平均応答時間が増加する関係を図4に示す。

## 3.2 測定ツールと予測ツール

以上がシングルプロファイル測定法の基本的な考え方とその背景であるが、この方法が成功するためには次の二つの条件が必要である。

- シングルプロファイルを簡単にかつ正確に測定できる測定ツールをもっていること、
- 待ち行列網モデルに基づき性能評価指標を計算する

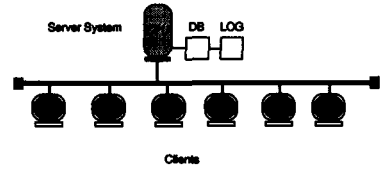


図1: システム構成例

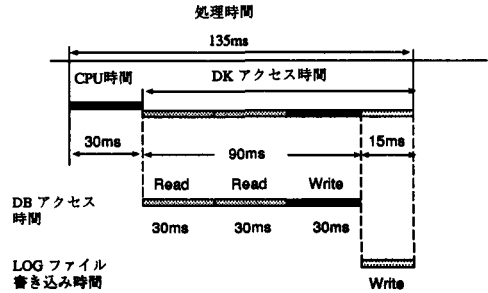


図2: シングルプロファイル例

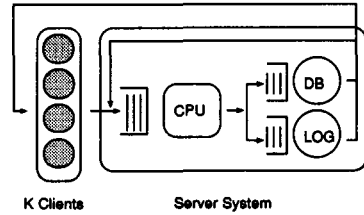


図3: 待ち行列網モデル

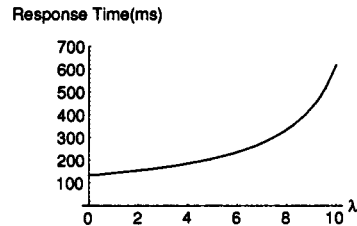


図4: 平均応答時間の予測例

表 1: シングルプロファイル分析

プロセス	CPU	DB DISK	Log DISK
AP		60ms (2 read )	
DB Writer		30ms (1 write)	
Log Writer			15ms (1 write)
total	30ms	90ms	15 ms

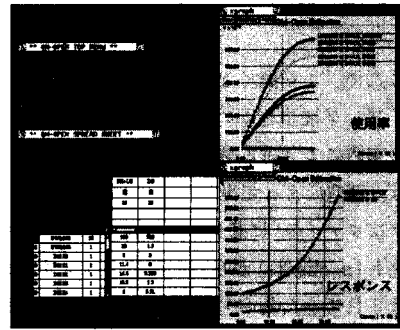


図 6: QM-Open の入出力例

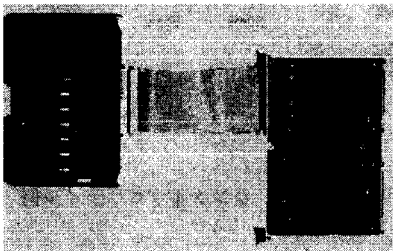


図 5: TinyTOPAZ の測定用ボード

計算ツールをもっていること、

この条件をみたす目的で開発された二つのツール、TinyTOPAZ と QM-Open についてその概要を以下に簡単に紹介する。

**TinyTOPAZ:** TinyTOPAZ はソフトウェア機能とハードウェア機能を併用して測定を行うハイブリッド型の測定ツールである。測定対象となるシステムの各種の事象をその発生時刻と共に記録して行くため、ソフトウェア機能として事象の発生とその種類を知らせるための仕掛けを OS のカーネル部分に埋め込む。この仕掛けのことをソフトウェアプローブという。ソフトウェアプローブにより検出された事象(イベント)はボードを介して電気信号として取り出され記録されて行く。図 5 に TinyTOPAZ の測定用のボードを示す。このようにして採取された記録を編集・解析する事によって様々なシステムの動きを精度良く知る事ができる。実際に TinyTOPAZ を用いて実測を行った結果の一例を図 8 に示す。横軸方向は測定時間の経過をしめし、左側には動作



図 7: QM-Open を使用しているところ

したプロセスの名前が表示される。また、それらのプロセス間の制御の移り変わりが矢印で示され、さらに各プロセス毎に CPU ならびに DISK 装置の使用時間と待ち時間が表示される。最上段には OS のカーネル部分が動作した時間が示されている。ハイブリッド型の測定ツールはハードウェア機能を利用するため測定のオーバーヘッドが極めて少ない事が大きな特長である。また、図 8 から分かるように、システムの動作を詳細に調べる事ができるので、パフォーマンスデバッガとしての利用方法にも十分に対応できる。

**QM-Open:** QM-Open は待ち行列網モデルを利用して各種の性能評価指標を算出するためのソフトウェアツールである。TinyTOPAZ により測定されたシングルプロファイルを入力データとして計算を行う。図 6 に QM-Open の入力および出力の例、図 7 に QM-Open を使用している場面を示す。待ち行列網モデルを利用し

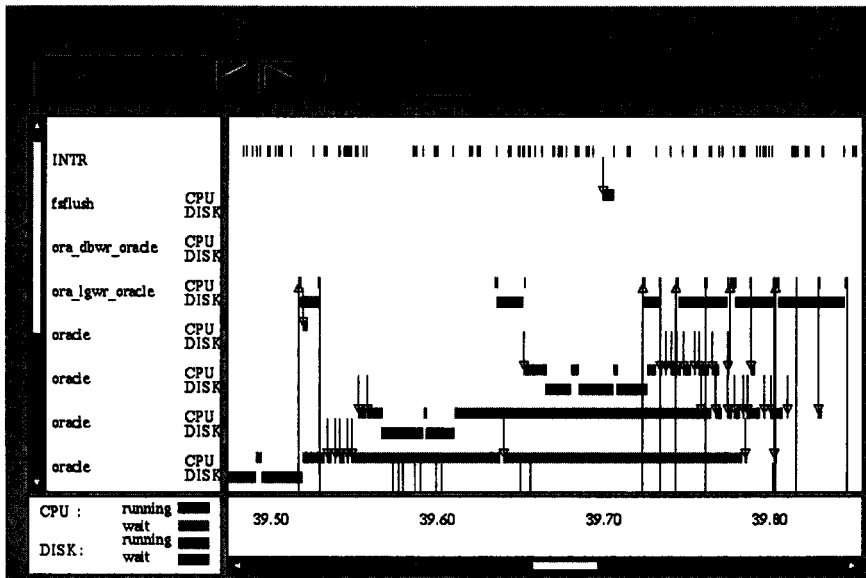


図 8: TinyTOPAZ による実測例

た計算により性能評価指標を算出するため、シミュレーションによる予測と比較して結果を得るまでの計算時間が極めて短くてすむ点が大きな特長である。また、表形式でシングルプロファイルデータを入力することによりモデル化が実行でき、プログラミングを伴うモデル化の作業が不要である事も大きな利点である。

### 3.3 シングルプロファイル法の検証例

シングルプロファイル法は先に述べたように、簡単にしかも正確な性能評価作業を行う事ができる方法である。しかし、各種の性能評価指標を予測する部分に待ち行列網モデルという数理的なモデルを利用しているため、そこに仮定されている各種の前提条件が果たして現実のシステムの特長を大過なく反映しているものなのかどうかを検証しておく必要がある。待ち行列網モデルを用いた予測は、負荷の増減とシステム性能指標の関係について、実際のシステム開発を数多く手掛けてきたシステム技術者達が経験的に獲得してきた知識と良く合うと言われてきてはいるが、その事を実験によりはっきりと確認しておくことは重要である。検証の一例として、図9に平均応答時間を実測値と予測値について比較したものを示す。この例は、金融関係のOLTPシステムを想定して実験したもので、横軸に示されるクライアント数の増加にともない応答時間が増加して行く様子がわかる。こ

の例では、平均誤差 10%、最大誤差 18%であった。本例以外の検証実験結果もほぼ同じ程度の誤差範囲を示していた。システム性能評価においては、この程度の誤差範囲で性能予測ができれば十分実用に耐え得るものと考えてよく、シングルプロファイル法の有効性は確認できたと考えられる。

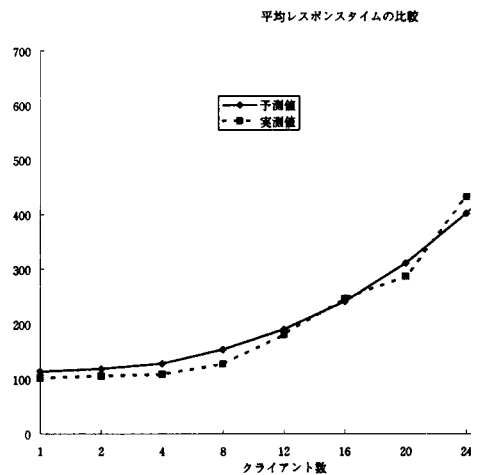


図 9: 予測値と実測値の比較

### 3.4 シングルプロファイルとトラフィック密度

シングルプロファイル法を支える重要な基礎技術の一つが待ち行列網モデルに関する技術である。システム性能評価に応用される待ち行列網モデル [3] は、積形式解をもつ網を基本形とし、実システムの特徴を反映させるために必要な近似解を付加して行く事が一般的に行われる。この分野の解説は紙面が尽きたので、別の機会にしたいが、ここでは測定で得られたシングルプロファイルと積形式解をもつ待ち行列網モデルの系の定常状態の同時分布に現れるトラフィック密度 の関係について簡単にふれておく。一般に、積形式解を定めるためには、客の網内移動経路を係数行列とする連立方程式を解き、その解として定まる(相対)訪問回数と各ノードのサービス率の比として定義されるトラフィック密度を定める必要がある。しかし、シングルプロファイル法では、1トランザクションあたりの各システム資源(ノード)の使用時間のみを入力情報として用い、資源間の移動に関する情報と各資源の1回当たりの使用時間に関する情報は使用しない。このため、入力情報は簡素化され、システム技術者が理解しやすい形に整理されている。この事は次の事実に着目することによって実現されている。即ち、積形式解を定めるために直接必要なものはトラフィック密度のみであり、客の網内移動経路情報とノードのサービス率についての情報は間接的なものである、という点である。トラフィック密度さえ同じならば客の網内移動経路が異なる網でも同じ積形式解をもつ。この事実を利用すれば、シングルプロファイル測定により得られた1トランザクション当たりのシステム資源の使用時間はそのまま対応するノードのトラフィック密度に置き換えて積形式解を定める事ができる。

## 4. おわりに

3回にわたり、情報処理システムの性能評価技術についての紹介を行った。システム性能評価技術は、性能測定に関する技術と性能予測に関する技術とから構成されている総合技術としてとらえられるべきものと思われる。性能評価技術に関しては、基本的な技術の枠組は殆んど出尽くしているとも言われているが、個々の技術領域においては解決すべき課題は山積しているように見える。この分野に踏み込んだ研究者は、性能評価にとって基本的とも思える問題がいまだに未解決であることをしばしば知る事になろう。一方、現実への応用の場面では、個々の技術領域で得られた成果を目的に向けうまく

組み合わせ利用して行くための様々な工夫と努力が必要になる。実際、「役に立つように見えるモデル」と「役に立つモデル」の間の距離は予想以上に大きいものである。このとき、個別領域の研究者に要求されているものは、現実の問題をよく観察し理解たうえでのモデル化と解析であり、その適応結果への誠実なフォローではないかと思う。一方、性能評価を実施するシステム技術者に要求されるものは、関連する技術領域の動向や成果をよく勉強し理解した上で、それらを現実問題に適用して行く応用力であろう。システム性能評価技術の進展とその応用面の広がりを促進するためには、両者が情報を交流できる適切な機会が必要であり、実際に本学会の「待ち行列研究部会」もこの点で大きな貢献をしてきていると言えようし、また今後の果たすべき役割への期待も大きい。

本稿では情報処理システムの性能評価に関する技術のうちの限られた内容しか取りあげる事ができなかったが、この分野に興味をもたれる方々にいささかなりとも参考になれば幸いである。

## 参考文献

- [1] 堀川 隆, 紀 一誠, DBMS 動作特性の測定・解析手法, 情報処理, 94-ARC-104-4/94-OS-62-4, 1994.
- [2] 堀川 隆, 田中 淳裕, 小林 和朝, 紀 一誠, オープンシステム向け性能評価ツール, 情報処理学会第 49 回全国大会 (6), pp.185-186, 1994.
- [3] 紀 一誠, 待ち行列ネットワークとその応用, 科研費シンポジウム報文集, 情報通信ネットワークに関する性能評価モデルの総合的研究, 研究代表者 橋田 温, 1995.
- [4] 田中 淳裕, 堀川 隆, 小林 和朝, 紀 一誠, オープンシステムにおける性能評価手法, 情報処理学会第 49 回全国大会 (6), pp.187-188, 1994.
- [5] A.Tanaka, T.Horikawa, K.Kobayashi and I.Kino, Capacity Planning for Client Server Database Systems: A Case Study, PERMEAN '95 (to appear).