計算機アーキテクチャ　104－4
システムソフトウェアと
オペレーティング・システム　62－4
（１９９４．　１．　２７）

# DBMS動作特性の測定・解析手法

堀川　隆、　紀　一誠

ＮＥＣ　Ｃ＆Ｃ研究所

本稿では、オープンシステムの性能評価手法として、シングル・プロファイル法を提案し、その有効性を検証する。シングル・プロファイルとは、無競合状態における各計算機資源の正確な使用時間であり、実システム動作状況の測定・解析により得る。提案手法は、このシングル・プロファイルを待ち行列網モデルに入力して性能予測を行なう方法である。

提案手法の有効性検証は、DBMSシステムを対象に行なった。すなわち、unixサーバ上でトランザクション処理を実行させたときの性能実測値と、提案手法による性能予測値を比較し、予測精度を調べた。この結果、シングル・プロファイルを採取したのと同じ構成のシステムはもとより、ディスク構成の異なるシステムの性能についても、充分な精度で予測できたことから、本手法は、オープンシステムの性能評価に有用であるとの結論を得た。

# Performance Evaluation Method for DBMS Systems

Takashi Horikawa, Issei Kino

C&C Research Laboratories,
NEC Corporation

In this paper, we propose the single profile method as a performance evaluation method for open systems, and confirm availability of this method. The single profile is a set of basic performance parameters that show accurate service times on system resources while a system processes one transaction under no resource contension. This profile is obtained by real system performance measurement with original hybrid-monitor. In the single profile method, we estimate system performance with queueing network model driven by the single profile.

To confirm availability of this method, we apply the method to unix server in processing a OLTP job. The estimated performance has compared with performance obtained by measurement. The comparison results show good agreement for not only a system from which a single profile was obtained, but also systems whose configuration are different from the measured system. Consequently, we concludes that the single profile method is suitable for open system performance evaluation.

## 1 はじめに

近年の計算機利用環境は、ダウンサイジング、オープン化、マルチメディア化の方向で進歩している。この傾向は、ビジネス分野においても例外ではなく、従来、汎用計算機で行なっていたデータベース処理をunixサーバに移し、クライアントサーバ型の処理を行わせる情報システムが普及し始めている。

オープン化されたunixサーバ上に情報システムを構築する場合、異なるメーカの開発した1)ハードウエアとOS、2)データベース管理システム(DBMS)、3)業務アプリケーション、を組み合わせることが多くなる。このような情報システムでは、関係するメーカ間で、互いに、他社開発部分の詳細情報は得られないことから、一般に、システム全体の動作を把握するのは困難である。

一方、性能評価は、情報システムの提案、開発、運用の各フェーズで重要な作業であるが、オープンシステムにおいては、上記の事情のため、実システムを構築して運用するまで、性能が判明しないことが多い。

シングルプロファイル法は、このようなオープンシステムを対象とする性能評価手法であり、実システム動作状況の測定・解析により得た無競合状態における各計算機資源の正確な使用時間を待ち行列網モデルに入力して性能予測を行なう。本稿では、この手法の概要を説明するとともに、DBMSシステムを対象にした性能評価結果を述べ、オープンシステム性能評価における提案手法の有効性を検証する。

## 2 性能評価手法

### 2.1 開発の方針

ここで評価対象としているクライアント・サーバ型のシステムのイメージを図1に示す。このようなシステムの性能は、クライアント・サーバ型システムの性能には、1)unixサーバ(以下、サーバ)、2)ネットワーク、3)クライアント、といった個々の要素性能が関係するが、本稿では、サーバの性能評価について述べる。一般に、サーバは、複数のクライアントからの処理要求(以下、トランザクション)を並行して処理することから、性能の評価が困難であるのに加え、クライアント・サーバ・システムの性能ボトルネックになる場合が多いためである。

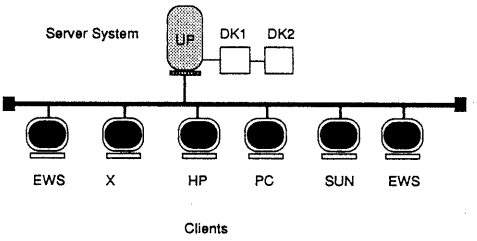


図1: クライアント・サーバ・システムの一例

サーバで処理すべきトランザクション数が1の場合、トランザクション処理時間(クライアントへのレスポンス時間)の見積り方法は単純であり、処理に要するプロセッサやディスクなどの計算機資源を使用する時間の合計となる。これに対し、複数のトランザクションを並行して処理する場合を考えると、処理に必要な計算機資源は、他のトランザクション処理で使用されている可能性がある、すなわち、トランザクション間で資源の競合が発生し、その結果として、資源待ち時間が生じることになる。従って、サーバにおけるトランザクションのレスポンス時間(性能)を見積もるには、資源待ち時間の評価が重要となる。

従来より、このような待ち時間の解析には待ち行列網モデルが多用されており、例えば、QM-X[1]は、BCMP型の待ち行列網モデルに基づく汎用計算機用の性能評価ツールとして実用化されている。本稿で述べるオープンシステム性能評価手法の開発においては、このような成果を継続発展させていくことを基本方針とした。

### 2.2 シングル・プロファイル法

シングル・プロファイルとは、無競合状態のサーバが、1トランザクションを処理するのに要したシステム資源使用時間である。このようなデータは、待ち行列網モデルによる性能予測に必須であり、従来は、汎用計算機に備わって

いる課金情報やシステム稼働状況のレポート、さらには、プログラム・リストからの推定により、処理に要するプロセッサ命令の数を得ていた。

unix サーバを中心とするオープンシステムにおいてシングル・プロファイルを得ようとすると、動作するプログラムの情報を得ることが困難なため、通常は、システム動作状況モニタ用のツール(例えば、sar)に頼る必要がある。しかし、sar は、下記の点で正確なシングル・プロファイルを得ることが困難であると考えた。

1. 時間の分解能は unix の管理するタイマに依存する。実際には、約 10m ～ 20mS の分解能しかない。
2. 数秒間の動作状況を集計したレポートが得られるのみであり、個々のトランザクション処理に要した計算機資源は判明しない。

シングル・プロファイルの正確さが、待ち行列網モデルによる性能予測の精度、信頼度を左右することから、提案手法では、新たに開発した測定ツール(2.3)を用いた実システムの測定によりシングル・プロファイルを得る。このように測定により得られた精密なシングル・プロファイルを、待ち行列網モデルに入力して性能予測を行なう手法を、ここでは、シングル・プロファイル法と呼ぶ。

## 2.3 性能測定ツール

unix サーバの性能基礎データは、独自に開発した測定ベースの計算機性能解析ツール TOPAZ [2] (Trace-data Oriented Processer AnalyZer) により得た。測定手法は、ソフトウエア・プローブとハードウエア・トレーサを併用するハイブリッド・モニタ方式[3]である。この方法の特徴は、1) 測定オーバーヘッドが小さいこと、2) ソフトウエア動作まで調べることが可能、の 2 点である。一般に、詳細なシステム動作を調べる場合、ソフトウエアのみから成るツールでは、測定時に発生するオーバーヘッドが 10 ～ 20% 以上になる点が問題となるが、本手法により、システムの動作を 1 ～ 2% の低オーバーヘッドで測定でき、正確なシングル・プロファイルを得ることが可能となった。

**ソフトウエア・プローブ** は、計算機動作を調べるのに必要な情報の検出を目的として、測定対象計算機のソフトウエアに埋め込む命令列である。ここでは、ソフトウエア・プローブを unix カーネル[4]内の関数に埋め込んで測定を行なった。測定対象としたのは、1) プロセスの状態変化などのソフトウエア実行状態の変化、2) 入出力アクセスの開始・終了、など、システム動作を調べるのに必要・不可欠なソフトウエア・イベントである。これらのイベントが発生すると、イベント ID およびシステム動作解析に必要な情報は、VME バス上に置かれた特定の出力ポートに書き込まれる。

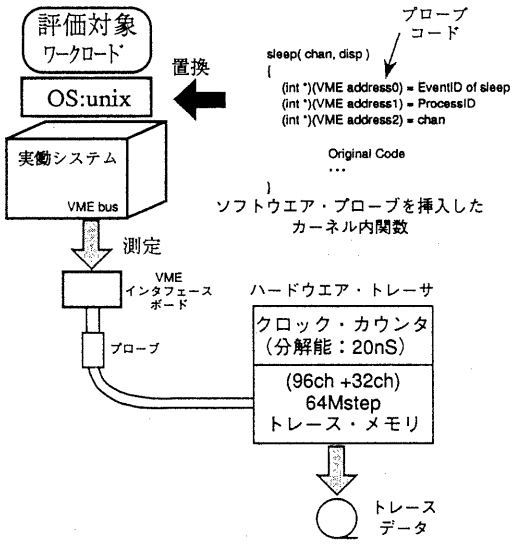


図 2: 測定系の全体像

**ハードウエア・トレーサ** はハードウエア・モニタの一種で、バスなどのハードウエア信号をモニタすることにより、計算機動作を調べる測定ツールである。一般に、ハードウエア・モニタは、測定オーバーヘッドが皆無である点が長所であるが、1) 採取可能なデータ量が少ない、2) ソフトウエア・イベントが検出できない、という問題点があるとされていた。ここでは、我々の独自開発した 64M ステップ・ハードウエア・トレーサを用いることにより 1) を解決した。また、ソフトウエア・プローブを併用する

ハイブリッド・モニタ方式の採用により、2) を解決した。

測定系の全体像を図2に示す。ハードウエア・トレーサでは、ソフトウエア・プローブによる書き込み信号をサンプル・クロックとして、アドレス・バスとデータ・バス上の値を採取し、これを時系列（トレース・データ）として記録する。

さらに、ハードウエア・トレーサでは、出力ポートに対して書き込みの行なわれた時刻、すなわち、ソフトウエア・イベントの発生時刻も記録する。これは、ハードウエア・トレーサのコンポーネントとして用意されているハードウエア・タイマを利用したものであり、その時間分解能は 20nS である。これにより、OS の機能としてソフトウエア的に実現されているタイマの分解能（～10mS）よりも5桁以上、細かい分解能でイベントの発生時刻を記録できる。

## 3 実働システムの測定

### 3.1 測定対象ワークロード

ここでは、DBMS に oracle V6 [5][6] を用いたシステムで、TPC-B ベンチマーク [7] を実行させたときの性能を評価した。

#### 3.1.1 トランザクション処理の概要

TPC-B は、1) 口座 (Account)、
2) 窓口 (Taller)、3) 支店 (Branch)、
4) 履歴 (History)、の4種類のデータベースに対して、下記のトランザクションを繰り返し発行 (クライアントの think time =0) したときの、スループット性能を測定するベンチマークである。なお、Accout, Taller, Branch のデータベースには、index を付与している。

| | |
|---|---|
| update A | 乱数で決る口座の残高に、乱数で決る取り引き額 (±がある) を加える |
| select A | 上記の処理を行なった後の口座残高額を得る |
| insert H | 履歴 DB に取り引き履歴を格納 |
| update T | 窓口残高に、取り引き額を加える |
| update B | 支店残高に、取り引き額を加える |
| commit | トランザクションを COMMIT |

#### 3.1.2 DBMS 動作の概要

oracle V6 によりトランザクション処理を実行させた時のデータの流れを、図3に示す。システム・グローバル領域 (SGA) は、データベースのディスク・キャッシュの役割を果たしており、アプリケーションの要求するデータが SGA に存在している場合は、ディスクをアクセスすることなくトランザクション処理を継続することができる。

従って、トランザクション処理において発生するディスク・アクセスは、主に、下記の3種類の要因によるものである。

1. アプリケーションの要求するデータが SGA に存在していない場合、shadow プロセスが DB ファイルを read する。
2. commit 時に、LGWR プロセスがログ・ファイルに write する。
3. トランザクション処理により書換えられた SGA 領域を DBWR プロセスがデータ・ベースに write する。

この内、最初の2つは、トランザクション処理と同期して実行されるが、最後の1つは、トランザクション処理とは非同期に実行される。

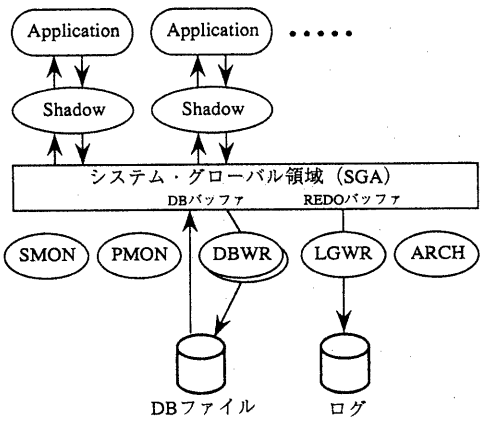


図 3: oracle V6 の動作

### 3.2 測定対象システム

測定対象システム (SUT) のハードウエア構成を図4に示す。SCSI バス (1本) に接続され

た5台のディスクは、各々、1) プログラム等の格納および swap、2)log1、3)log2、4)account data 格納用、5) その他の data base、および、account index 格納用、として使用した。

なお、log 用に 2 台のディスクを用意しているが、システムの稼働状態では、アクセスされるのはどちらか一方であるため、図 4に示す SUT で TPC-B を実行させた場合、アクセスされるディスクは 4 台である。

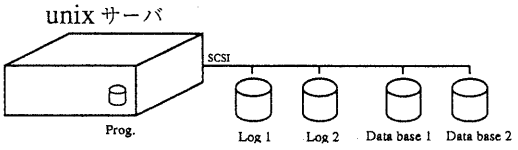


図 4: 測定対象マシン (SUT) の構成

### 3.3 シングル・プロファイルの解析

oracle V6 では、主にトランザクション処理に携わるプロセスは、下記の 4 種類である。

| | |
|---|---|
| Application | トランザクションを発行するプロセス |
| Shadow | アプリケーション・プロセスと 1 対 1 に対応する |
| LGWR | トランザクションの commit 時に、ログをディスクに書き込む |
| DBWR | SGA 上で更新された DB のデータをディスクに書き戻す |

そこで、これらのプロセスが、ベンチマーク実行期間全体を通して使用した CPU 時間、ディスクアクセス時間の合計を、実行したトランザクション数で割ることにより、シングル・プロファイルを求めた。なお、測定時間、すなわち、プログラム実行時間は、ログのチェック・ポイントによる性能の落ち込みを含むように、約 10 分とした。

1 トランザクションの処理時間を 1.0 として正規化したシングル・プロファイルを表 1、表 2に示す。この結果は、アプリケーション・プロセス数が 1 の場合の測定結果から求めたものである。なお、application プロセスは、ディスク・アクセスを行なわなかったため、表 2には記載していない。

| Application | Shadow | LGWR | DBWR |
|---|---|---|---|
| 0.07 | 0.25 | 0.03 | 0.02 |

表 1: プロセッサ時間 [相対値]

| アクセス対象 | Shadow | LGWR | DBWR |
|---|---|---|---|
| Tbl-A | 0.23 | - | 0.21 |
| Tbl-T | - | - | 0.0 |
| Tbl-B | - | - | 0.0 |
| Idx-A | 0.17 | - | - |
| Idx-T | - | - | - |
| Idx-B | - | - | - |
| History | - | - | 0.01 |
| その他 | 0.0 | - | 0.01 |
| Log | - | 0.15 | - |
| Prog. | - | 0.0 | - |

表 2: ディスク・アクセス時間 [相対値]

表 2において、アクセス対象は、次の通りに分類した。また、測定対象システムでは、横の罫線で区切られた範囲のデータを、1 台のディスクに格納している。

| | |
|---|---|
| Tbl-x | 各 DB のデータ (x=A:Account, B:Branch, T:Taller) |
| Idx-x | 各 DB のインデックス (同上) |
| History | 履歴 DB のデータ |
| その他 | ロールバック・セグメントや、DB のシステム領域 |
| Log | ログ (log1, log2 の内、ベンチマーク実行時に使用していた log 用ディスク) |
| Prog. | プログラム、swap 領域 |

## 4 性能評価

アプリケーション・プロセス数 (以下、クライアント数) を変えることにより DBMS への負荷を調節すると、システムのスループットが変化する。図 4に示すシステムでの、クライアント数が 1 の場合の性能実測値を基準 (1.0) と

し、クライアント数増加により、性能が基準性能の何倍になるかを待ち行列網モデルによって予測した結果と実測値と比較する。これにより、提案手法の有効性を検証する。

### 4.1 待ち行列網モデルによる性能予測

#### 4.1.1 性能予測モデル

BCMP 型待ち行列に基づく評価モデルを作成して、性能予測を行なった。このモデルは、プロセッサとディスクを計算機資源として表現したセントラル・サーバ・モデルであり、プロセッサとディスクに関する競合を評価できる。

BCMP 型待ち行列網モデルは、互いに非同期に動く複数の連鎖を扱える点が特徴である。ここでは、下に示す 2 種類の連鎖により、システム動作をモデル化した。

1. Application, shadow, LGWR プロセス
   (oracle V6 によるトランザクション処理では、これらがのプロセスがシリアルに動作する。)
2. DBWR
   (このプロセスは、上記のトランザクション処理とは非同期に動作する。)

また、プロセッサに対して定期的に発生するクロック割り込み処理の影響も考慮した。

#### 4.1.2 性能予測結果

前節で述べた待ち行列網モデルに、3.3で得たシングル・プロファイルを入力し、図 4のシステムについて性能予測を行なった。このプロファイルによると、Tbl-A の使用量が最も多いことから、このシステムは、Tbl-A のディスクが性能のボトルネックになると考えられる。

この結果と実測値との比較を図 5に示す。

測定によると、このシステムの性能は下記の傾向を示した。

- クライアント数が 8 程度までは、スループット性能が徐々に向上する。
- クライアント数が 8 ～ 10 以上の領域では、クライアント数 1 の場合の約 2.2 倍のスループット性能で飽和する。

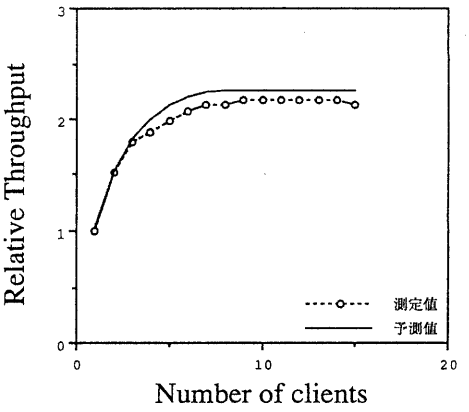


図 5: 性能予測値と測定値の比較

測定結果と、4.1.1で説明した評価モデルによる予測結果は、クライアント数が 3 以下の領域で、ほぼ一致し、それ以上の領域でも、7.5% 以下の誤差で一致した。

### 4.2 構成の異なるシステムの性能予測

#### 4.2.1 全 DB を同一ディスクに置くシステム

Tbl-A とその他の DB を同一のディスクに格納したシステムについて、スループット性能の実測を行ない、性能予測値との比較を行なった。このシステムでは、計算機資源はプロセッサとディスク 3 台となる。これらのディスクに格納するデータは、次の通りである。

| ディスク | 格納するデータ (表 2参照) |
|---|---|
| Disk 1 | Tbl-A, Tbl-B, Tbl-T, Idx-A, Idx-B, Idx-T, History, その他 |
| Disk 2 | Log |
| Disk 3 | Prog. |

性能評価に用いるシングル・プロファイルは、新たに測定して得るのではなく、3.3に示したプロファイル、すなわち、評価対象システムとはディスク構成が異なるシステムのプロファイルから作成した。両者のプロファイルが異なるのは、表 2において、Tbl-A を格納するディスクとその他の DB を格納するディスクの

合計が全DBを格納するディスクの使用量となる点である。

待ち行列網モデルによる予測結果と実測値との比較を図6に示す。なお、前節に示した結果との比較のため、本図も、図4に示すシステムでの、クライアント数が1の場合の性能実測値を基準(1.0)として表示した。

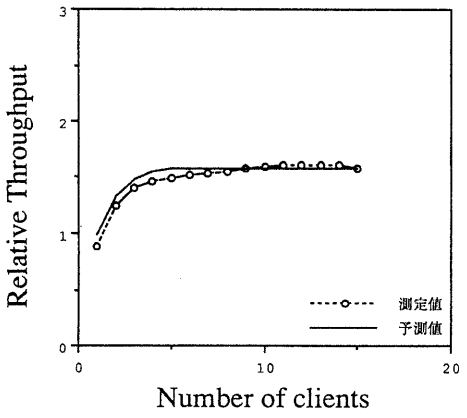


図 6: 性能予測値と測定値の比較

このシステムでは、ディスクがシステム性能の極端なボトルネックになるため、スループット性能はクライアント数が6程度で飽和するという測定結果が得られた。このシステムの場合、評価モデルによる予測結果は、クライアント数の全域(1～15)について、11%以下の誤差で一致した。

### 4.2.2 Account DBを分割したシステム

DBの中で最もアクセス頻度の高いaccount dataを2台のディスクに分割して格納するシステムについて、スループット性能の実測を行ない、性能予測値との比較を行なった。このシステムでは、計算機資源はプロセッサとディスク5台となる。これらのディスクに格納するデータは、次の通りである。

| ディスク | 格納するデータ(表2参照) |
|---|---|
| Disk 1 | Tbl-A の 1/2 |
| Disk 2 | Tbl-A の 1/2 |
| Disk 3 | Tbl-B, Tbl-T, Idx-A, Idx-B, Idx-T, History, その他 |
| Disk 4 | Log |
| Disk 5 | Prog. |

このシステムのシングル・プロファイルは、前節と同様、3.3に示したものから作成した。両者が異なっているのは、表2において、Tbl-Aを格納するディスクの使用量を1/2倍した値を、このシステムにおいて、Tbl-Aを格納するディスク2台の使用量とした点である。作成されたプロファイルによると、プロセッサの使用量が最も多くなることから、本システムの性能ボトルネックは、プロセッサになると考えられる。

待ち行列網モデルによる予測結果と実測値との比較を図7に示す。前節と同様、本図も、図4に示すシステムでの、クライアント数が1の場合の性能実測値を基準(1.0)として表示した。

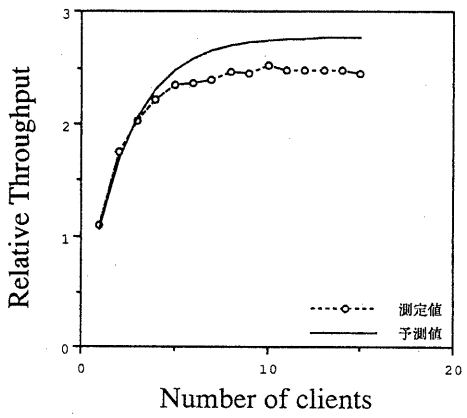


図 7: 性能予測値と測定値の比較

このシステムの場合、クライアント数が4以下の領域では、測定値と予測値がほぼ一致したが、それ以上の領域では、10～15%以下の精度での一致となり、4.1.2よりは、若干、誤差が大きくなった。この理由については、4.3で

考察する。

### 4.3 考察

4.1.2および4.2に示した結果より、シングル・プロファイル法によると、プロファイルを採取したシステムだけでなく、ディスク構成の異なるシステムについても、充分実用になる精度で性能を予測できることがわかった。

また、プロセッサが性能のボトルネックと考えられるシステムでは、クライアント数が増加したときの予測性能の誤差が、若干大きくなった。この原因は、まだ明確になってはいないが、下記のいずれかの要因によるものであろうと推察している。

1. シングル・プロファイルにおいて、プロセッサ時間として計上しなかったシステムdaemonの類がプロセッサを使用していたため、トランザクション処理で使用できるプロセッサ能力が実質的に減少していた。
2. プロセッサ以外の要因が性能ボトルネックになっていた。例えば、データベースの一貫性を保証するために必要なロック等の論理的な資源である。

なお、このような実測と予測の不一致が生じた場合でも、正確で詳細なシステム動作を把握することにより、詳細な原因の追究、さらには、評価モデルの改良が可能であると考えている。

## 5 まとめ

本稿では、オープンシステムにも適用可能な性能評価手法としてシングル・プロファイル法を提案し、DBMSの性能評価に適用した結果を述べた。

本手法は、システムへの負荷が増大したときのシステム性能、さらには、測定したシステムとはディスク構成の異なるシステムの性能も精度良く予測可能であることが判った。

今後は、性能評価モデルを拡張し、種々のトランザクションを処理する場合の性能を評価できるようにするとともに、評価精度の更なる確認を行なっていく予定である。


## 謝辞

待ち行列網モデルによる性能評価モデルについて貴重な御意見を頂きましたNEC C&C研究所 小林和朝課長に深く感謝します。また、DBMSシステムの測定に協力頂きました田中淳裕氏、ならびに、性能評価モデル作成と性能予測計算を行なって頂きました高橋勇人氏、また、測定データ解析用ソフトウエアを開発して頂きましたNEC技術情報システム開発(株)加藤哲主任、吉田浩司氏、石井直子さんに感謝します。


## 参考文献


[1] 紀一誠、納富研造、待ち行列網モデルによる計算機システムの性能評価用ソフトウエアパッケージQM-X、情報処理学会論文誌、Vol.25, No.4, pp.570-578, 1984.

[2] T.Horikawa, TOPAZ: Hardware-Tracer Based Computer Performance Measurement and Evaluation System, NEC R&D, Vol.33, No.4, pp.638-647, 1992.

[3] P.McKerrow, Performance Measurement of Computer Systems, Addison-Wesley, 1987.

[4] M.J.Bach, The Design of The UNIX operating system, Englewood Cliffs, NJ: Prentice-Hall, 1986.

[5] 日本電気株式会社, ORACLE導入の手引き (WKD50-2),1992年12月

[6] 日本電気株式会社, ORACLEデータベース管理の手引き (WKD51-1),1991年9月

[7] Jim Gray: The Benchmark Handbook for Database and Transaction Processing Systems, (Morgan Kaufmann Publishers, 1991). 喜連川, 渡辺訳: データベース・ベンチマーキング, 日経BP社(1992).