

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ェア・パッケージ QM-X†

紀 一 誠†† 納 富 研 造†††

本稿は混合型 BCMP 待ち行列網を用いた計算機システムのモデル化に関する考え方と、それを具現化した性能評価用ソフトウェア・パッケージ QM-X について述べる。複雑化、大規模化を続ける計算機システムの性能評価を短期間に少ない工数で的確に行うためには、明確な方法論とその実現のための簡便なツール類が完備していることが必要である。積形式解をもつ待ち行列網、とりわけ BCMP 型待ち行列網はその一つの有効なモデルとして広く研究され、これを基礎とするパッケージの開発も数多く報告されている。BCMP 網の性能評価モデルへの応用に際してはおのおの独自の工夫がなされているが、本稿で示す方法は次の点を特長としている。混合型 BCMP 網に関する計算法を開発したことにより、従来以上に柔軟なモデル化を可能にしたこと。プロセスのアイドル状態を表現する遅延ノードの概念を導入し、このモデルの数値解法を開発することにより、閉鎖型サブシステムにおいてもジョブの到着件数を利用者が直接指定できるようにしたこと。系の状態確率を示す積形式表現のなかに推移確率行列の要素が陽に現れないことを利用し、入力情報を必要最小限の理解しやすいものとしたこと。状態確率が積形式表現にはならないプロセス待ち行列についての近似解を与えることにより、プロセス待ち時間の近似計算を可能にしたことなどである。

1. はじめに

計算機システムと通信網を基礎として構築される情報処理システムは、利用形態においても応用目的においても近年ますます多様化するとともに、大規模で複雑なものになりつつある。大規模で複雑なシステムをその利用目的に沿うよう最適に設計するためには、システム機能に関する定性的な設計とともに、システム性能に関する定量的な評価を両輪として進まねばならない。さらに、運用後のタイミングのよい機器増設や機種種の切換え、新規業務の追加等を的確に行うためにもシステム性能の定量的評価は欠かせない。性能評価を誤ったり、あるいは怠ったりした場合、これにより発生する損失費用は一般に、性能評価作業に必要な費用よりも桁違いに膨大なものになる。したがって、性能評価作業はシステムのライフサイクルのなかのいかなる段階でも注意深く実行されねばならない基本的な作業であり、このための技術はシステム技術の基礎をなすものである。しかし、性能評価を実施するシステム部門でも年々作業効率の改善と生産性の向上が追求され、システム技術者たちは日を追ってより少ない工数でより質の高い作業結果を出すことが要求されている。

少工数、短期間に的確な性能評価を実行するためには次の要件が欠かせない。

- 1) 方法論が明確であり、標準化されていること。
- 2) この方法論に沿って作業を進めるための強力なソフトウェア工具が豊富に準備され手軽に使えること。

この要件を満たす一つの技法として、ここ数年の待ち行列網理論およびそのシステム性能評価への応用の発展はめざましいものがある。

閉鎖型の Jackson 網を利用したセントラルサーバモデルに始まる待ち行列網モデルの応用は、1975年に積形式をもつ待ち行列網の範囲が Baskett et al.¹⁾により大きく拡大され (BCMP 網といわれる)、Reiser and Kobayashi²⁾により同年閉鎖型 BCMP 網の計算方法が開発されたことによって飛躍的に応用範囲が拡大された。

さらに、この BCMP 型待ち行列網に基礎を置く性能評価用のソフトウェア・パッケージの開発も数多く報告されており、おもなものとして、QNET⁴⁾、BEST/1⁴⁾、QSEC⁵⁾、PNET⁶⁾、QNAP⁷⁾、QM-X⁸⁾等があげられる。使用目的は商用のものから研究用まで多岐にわたり、それぞれに応用に際しての独自の工夫がこらされており特徴をもっている。このうちの一つである QM-X は、文献 9) に示される混合型 BCMP 網に関する計算方法に基づき、梗概に示されるような他のツールとは異なるいくつかの特長を実現したソフトウェア・パッケージである。

また、実測値と待ち行列網モデルによる評価結果と

† QM-X: A Performance Evaluation Software Package Based on BCMP Queueing Networks by ISSEI KINO (C & C Systems Res. Labs., NEC Corporation) and KENZŌ NOUDOMI (EDP Marketing Support Division, NEC Corporation).

†† 日本電気(株) C & C システム研究所

††† 日本電気(株)情報処理営業支援本部

の比較検討も数多く報告されており、文献 10)~12) に詳しい。

本稿は、この QM-X に即して、BCMP 型待ち行列網モデルによる計算機システムの性能評価の考え方とその実現方法について述べる。2 章に設計目標、3 章に計算機システムのモデル化の方法について述べる。4 章および 5 章に QM-X の内部機構および機能概要を示し、QM-X の独自の考え方や応用上の工夫について述べる。6 章には簡単な使用例を示す。

2. QM-X の設計目標

QM-X は次の設計目標に基づいて作成された、BCMP 型待ち行列網に基礎をおく、情報処理システムの性能評価のための汎用ソフトウェア・パッケージである。

- 1) 性能評価作業の経験のないシステム技術者でも簡単に利用でき、質の高い結果が出せること。
- 2) 待ち行列網に関する用語や概念あるいは数学的な詳細に関する予備知識がなくても、計算機システムの世界の用語や概念だけを用いてモデルが作れること。
- 3) バッチ処理、リアルタイム処理、TSS 等の性格の異なるサブシステムが一つのシステム内に混在して動く多次元処理システムの性能評価が簡単に実行できること。
- 4) 対話型の処理を基本とし、パラメータの修正、モデルの変更が容易に実行でき、迅速な評価が繰り返せること。
- 5) 入力情報を必要最少限に精選し、豊富な出力情報編集機能をもつことにより、操作性のよいものにする。

3. QM-X における計算機システムのモデル化

本章では待ち行列網を用いた計算機システムのモデル化の考え方を QM-X に即して示す。

3.1 システム構成のモデル化

システムは一般に、競争を発生する N 個のシステム資源から構成されると考える。これらのシステム資源は待ち行列網を構成するノードに対応づけられ、実システムでは競争を発生する CPU や各種の入出力装置等に対応づけられモデル化される。各システム資源に対応するノードは複数窓口であってよい。資源の 1 回当りの使用時間は指数分布に従うものとし、資源を

利用希望者に割り当てるサービス規律は先着順とする。

このことは、資源は複数窓口のタイプ 1 (文献 1) の意味で) のノードでモデル化されることを意味している。

3.2 サブシステムのモデル化

BCMP 網を利用すると、一つのシステムのなかにバッチ処理サブシステム、リアルタイム処理サブシステム、TSS 等の性格の異なる M 種類のサブシステムが混在する多次元処理システムのモデル化を行うことができる。各サブシステムは、待ち行列網中の客の移動経路を確率的に指定する部分連鎖 (subchains) が対応づけられる。部分連鎖には、網外に客の出入のある開放型と、つねに一定人数の客が網内を移動する閉鎖型のもの 2 種類ある。おのおのに対応して、サブシステムは閉鎖型あるいは開放型サブシステムに類別される。QM-X ではこの両者の混在を許している。すなわち、QM-X は混合型の待ち行列網を基礎としている。

3.3 ジョブのモデル化

利用者がシステムに要求する処理単位の名称は、バッチ処理ではジョブ、リアルタイム処理ではトランザクション、TSS ではコマンド等とサブシステムごとに異なっている。本稿では簡単のため、これらの処理単位をすべてジョブと表現することにする。ジョブはシステム資源を次々と使用しながらその処理を進めていく。異なるサブシステムに属するジョブは、一般には、確率的に異なるふるまいと特性をもっている。

サブシステム l に属するジョブが、その処理の開始時点から処理終了時点までの間に、システム資源 i を使用する総回数を θ_{il} 回とし、システム資源 i の 1 回当りの平均使用時間を h_i 、 $\rho_{il} = \theta_{il} h_i$ とする。 ρ_{il} はジョブが使用する資源 i の総使用時間を意味している。ベクトル $\mathbf{r}_l = (\rho_{l1}, \rho_{l2}, \dots, \rho_{lN})$ をサブシステム l のジョブ特性と表現することにする。さらに、同じサブシステム l に属するジョブでも異なる特性をもつものが J_l 種類存在する場合を考える。サブシステム l に属する j 種類目のジョブが資源 i を使用する総回数を θ_{ij} 回とし、 $\rho_{ij} = \theta_{ij} h_i$ 、 $\mathbf{r}_{lj} = (\rho_{lj1}, \rho_{lj2}, \dots, \rho_{ljN})$ とする。また、サブシステム l に属するジョブ j の相対発生率を φ_{lj} とする。ただし、 $\sum_{j=1}^{J_l} \varphi_{lj} = 1$ 。

ベクトル \mathbf{r}_{lj} および $\{\varphi_{lj}\}$ 、 $j=1, 2, \dots, J_l$ をサブシステム l に関するワークロード特性ということにす

る。QM-X では、このジョブ特性あるいはワークロード特性を与えることによりジョブのモデル化を行う。

3.4 プロセスのモデル化

実システムでは一般に、オペレーティングシステム(OS)がシステム資源を割り付け制御する対象の単位として、プロセス(あるいはタスク)が定義されている。OSはジョブが到着するとそのジョブにプロセスの一つを割りあてる。OSはプロセスに対して、CPUをはじめとする各種のシステム資源の割り付け管理を実行する。したがって、ジョブが処理されるためにはその実行に先立ってプロセスが確保されていなければならない。この意味で、プロセスはシステムに用意されている資源の一種であると考えることができる。QM-Xではプロセスを次のようにモデル化している。

サブシステム l にはプロセスといわれる K_l 個の資源が用意されている。 K_l が有限の値をとる場合には、サブシステム l は閉鎖型であるといい、 K_l が無限の値をとる場合には開放型であるという。プロセスにはアクティブおよびアイドルとよばれる二つの状態が定義されている。ジョブがシステムに到着するとOSはアイドル状態にあるプロセスのなかの一つを割り付ける。

ジョブはプロセスが確保できた場合に限ってシステム資源の使用が可能となり処理が開始される。このとき、プロセスの状態はアイドルからアクティブに変化する。ジョブの処理が完了するとジョブはプロセスを解放し、プロセスの状態はアクティブからアイドル状態に変化する。したがって、プロセスがアクティブである状態とはジョブ処理中、すなわちジョブがシステム資源を使用中あるいは待合せ中の状態を表現し、アイドル状態はそのプロセスが処理すべきジョブが存在しない状態を表現している。ジョブの到着時点でプロセスに空きがない場合、すなわちアイドル状態のプロセスが存在しない場合には、そのジョブはプロセスに空きができるまで待ち合わせる。この状態にあるジョブの待ち時間をプロセス待ち時間という。開放型サブシステムではプロセスが無限に用意されているのでプロセス待ち時間は発生しない。

サブシステムごとのプロセス数 K_l は、待ち行列網の中を部分連鎖 l に従い網内を移動する網内容数に対応づけられている。また、アクティブ状態にあるプロセス数は、プロセスの実行多重度、あるいはマルチプログラミング多重度に対応づけられている。

表 1 負荷条件の指定方法
Table 1 Conditions for job arrival.

指定法	到着件数	最大処理件数	ジョブ空き時間
サブシステム			
閉鎖型	可	可	可
開放型	可	不可	不可

3.5 負荷条件のモデル化

QM-Xでは、各サブシステムごとに、負荷条件を次の3通りのいずれかの方法によって指定することが可能である。

1) 到着件数指定

サブシステム l に属するジョブが単位時間当りに到着する件数 λ_l (件/時間)を直接指定する。系の定常状態においては、到着件数=処理件数、であるから λ_l はまた、サブシステム l に属するジョブの単位時間当りの処理件数を示している。指定された λ_l がシステムの処理能力の範囲に入っているかどうかを判定し、処理能力を越えている場合にはその旨を通知し再指定を待つ。

2) 最大処理件数指定

サブシステム l に属するジョブに関して、システムが処理可能な最大の処理件数を算出するように指示するものである。ただし、このときの最大処理件数は、他のサブシステムの負荷の状況に影響されるので、他のサブシステムの負荷条件が与えられたという条件付きでの最大処理件数の意味である。

3) ジョブ空き時間指定

サブシステム l に属するプロセスのアイドル時間 u_l を直接指定する。 u_l は一つのプロセスからみて、ジョブの終了時点から次のジョブの到着までの平均時間間隔を表現している。

以上の3種類の負荷条件の指定方法とサブシステムのタイプとの関係を表1に示す。

4. QM-X の内部機構

本章では前章に示したモデル化を可能にするためのQM-Xの内部機構とその特長について述べる。

4.1 待ち行列網

文献1)では、開放型、閉鎖型の混在する複数個の部分連鎖、先着順、プロセッサシェアリング、無限サーバ、後着順割込優先中断点再開の4種類のサービス規律、複数種類の客のサービスクラスと広い範囲(Coxian)のサービス時間分布をもつような待ち行列網の定常状態確率が積形式に表現できることが示され

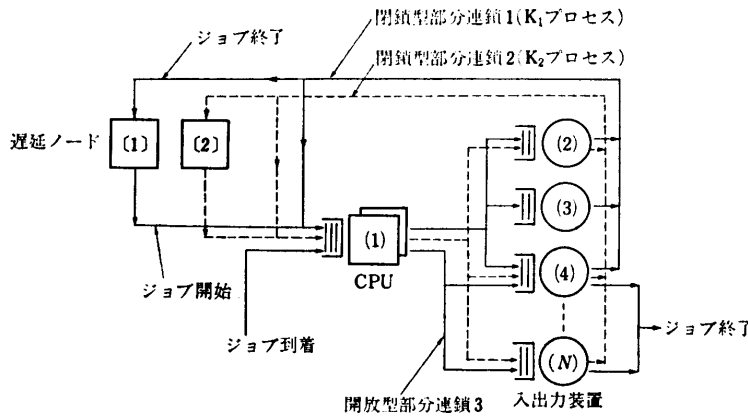


図1 二つの閉鎖型サブシステムと一つの開放型サブシステムをもつモデルの例

Fig. 1 An example of system model with two closed subsystems and one open subsystem.

ており、この種の待ち行列網は BCMP 型待ち行列網といわれている。

QM-X は BCMP 網を基礎としているが、計算機システムのモデル化と評価を目的とするため、BCMP 網で許されるすべての条件をそのままの形で利用者が指定選択するネットワーク・ソルバの形態を採るのではなく、計算機システムのモデルになじみやすくするために次のような待ち行列網を利用している。

1) ノードはすべて文献1)の意味で可変サービス率をもつタイプ1とする。すなわち、先着順サービスでサービス時間は指数分布に従うものとする。したがって、サービス時間分布の相違を反映する客のサービスクラスの相違に関する概念はモデルには採用されていない。

2) 複数個の閉鎖型と複数個の開放型部分連鎖が混在する混合型の待ち行列網を基礎とする。簡単のため、1番から L 番までを閉鎖型部分連鎖、 $L+1$ から M 番までを開放型部分連鎖とする。おのおのの開放型部分連鎖は、対応する客のポアソン発生源を網の外部にもつものとする。また、閉鎖型部分連鎖 l に従う網内客数を K_l とし、客数ベクトルを $\mathbf{K}=(K_1, K_2, \dots, K_L)$ とする。

3) プロセスがアイドルである状態を表現するために、おのおのの閉鎖型部分連鎖 l に対応して窓口数 K_l をもつ仮想的なノード(タイプ1)を一つずつ設ける。このノードを遅延ノードとよぶことにする。遅延ノードは網内客数に等しい窓口数をもつために待合せは発生せず、たんに時間の経過のみを表現することになる。

図1に、二つの閉鎖型サブシステムと一つの開放型サブシステムをもつモデルの例を示す。

4.2 システム資源への負荷を表すベクトル

QM-X では、3.3 節に示したように、ジョブ特性 r_i あるいはワークロード特性 r_{ij} , $\{\varphi_{ij}\}$ を入力パラメータとしている。

ワークロード特性が入力された場合には次のように φ_{ij} で加重平均をとることによりジョブ特性に換算する。

$$\rho_{il} = \sum_{j=1}^{J_l} \varphi_{ij} \rho_{ij},$$

$$\mathbf{r}_l = (\rho_{l1}, \rho_{l2}, \dots, \rho_{lN})$$

$l=1, 2, \dots, M$ について \mathbf{r}_l が得られたら、次のようにしてシステム資源 i に対する負荷ベクトルを構成する。

$$\rho_i = (\rho_{i1}, \rho_{i2}, \dots, \rho_{iM}), \quad i=1, 2, \dots, N$$

一般に待ち行列網モデルでは、客の網内移動経路を確率的に指定する推移確率行列を係数とする連立方程式を解いて、その解として定まる平均訪問回数ベクトルを用いてノード i への負荷ベクトルを定める。しかし、この手続きを忠実に実行しようとする、推移確率行列の要素 p_{ij} を入力としなければならない。 p_{ij} は実システムの動きと対応づけのむずかしいパラメータであり、また入力個数も最大 $N \times N$ 個必要なため、推移確率行列を入力情報に選ぶのは得策ではない。QM-X ではこの問題点を避けるため、待ち行列網の定常状態確率を示す積形式表現のなかには推移確率行列の要素は陽には含まれず、連立方程式の解として定まる $\{\theta_{ij}\}$ のみが陽に現われることを利用して、推移確率行列を入力情報とする代りにジョブ特性、すなわち負荷ベクトルを直接入力情報とする方法を採用することによって、過剰情報の入力を避け、入力情報を大幅に簡素化しわかりやすいものとした。

4.3 待ち行列網の計算法

QM-X では、混合型の多次元処理システムの性能評価を実現するため、先に開発された混合型 BCMP 網に関するたたみこみ計算法⁹⁾を計算の基礎としている。

4.4 遅延ノード

遅延ノードを設定することにより、プロセスがアイドル状態にある時間を表現することができ、閉鎖型サブシステムの場合にも到着件数 λ_i を陽に指定するこ

とが可能になる。閉鎖型サブシステム l に対応する遅延ノードの番号を $N+l$ ($l=1, 2, \dots, L$) とする。遅延ノード $N+l$ に関する負荷ベクトルは、 $\rho_{N+l}=(0, \dots, 0, u_l, 0, \dots, 0)$ と l 番目の要素のみ u_l でその他は 0 であるベクトルとなる。遅延ノードを含めて、すべてのノードに関する負荷ベクトル $\rho_i, i=1, 2, \dots, N+L$, が得られればこの待ち行列網に関する計算が実行でき、さまざまな性能評価指標を導くことができる。サブシステム l に関する負荷条件の指定方法が最大処理件数指定あるいはジョブ空き時間指定の場合には u_l は直接指定されることになるので負荷ベクトル ρ_{N+l} はただちに定まる。しかし、処理件数指定の場合には u_l が直接指定されるのではなく、到着件数 λ_l が指定されるので、負荷ベクトルはただちには定まらない。したがって、与えられた λ_l から u_l を計算する方法が必要とされる。以下この方法についての概要を示す。

サブシステム l に属するジョブの処理時間を $E_l(u_l)$ とすると次の関係が成り立っている。

$$\lambda_l = K_l / \{E_l(u_l) + u_l\} \quad (4.1)$$

上式は u_l に関しては K_l 次の代数方程式を意味しており、一般には解析解は得られず、何らかの数値解法に頼らざるを得ない。QM-X では、(4.1)式から次の計算スキームを導き、数値計算により与えられた λ_l から求める u_l を計算する方法を採っている。 $u_l(n)$ を n ステップめの近似値とし、初期値は $u_l(0) = 0$ とする。

$$u_l(n) = K_l / \lambda_l - E_l(u_l(n-1)) \quad (4.2)$$

(4.2)式に基づき、必要な精度が得られるまで n に関する繰返し計算を行う。一般には、負荷条件として到着件数を指定するサブシステムは複数個存在するので、(4.2)式をベクトル化し計算することになる。

QM-X では収束判定条件として次を採用している。

$$\{E_l(u_l(n)) - E_l(u_l(n-1))\} / E_l(u_l(n-1)) < 0.01 \quad (4.3)$$

実際の使用例では、(4.3)の判定条件のもとで、ほとんどが 3~4 回以内の反復回数で収束した。しかし、収束条件および収束の速さに関する厳密な考察は今後の課題として残されている。

4.5 プロセス待ち時間の計算

閉鎖型サブシステムに属するジョブが到着時に空きプロセスがない場合にはプロセス待ち行列中で待ち合わせる。このプロセス待ち行列を含んだ待ち行列網は定常状態確率が積形式にはならないことが知られてい

るので、プロセス待ち時間を計算するためには何らかの近似的なモデル化が必要とされる。QM-X では次のような考え方でプロセス待ち時間を計算している。

サブシステム l の遅延ノードには K_l 個のプロセスが用意されているので、これを窓口数 K_l をもつ $M/M/K_l$ 待ち行列とみなし近似を行う。プロセスがアクティブ状態にある時間、すなわちジョブ処理時間 $E_l(u_l)$ を窓口の使用時間とみなし、到着率は λ_l そのものとする。このとき、 K_l の窓口には、 $A_l = \lambda_l E_l(u_l)$ (アーランのトラヒックがかかっていると解釈される。 λ_l が与えられると前節の方法で u_l および $E_l(u_l)$ を計算することができ、 A_l を求めることができる。この A_l を用いて $M/M/K_l$ の待ち時間 $W_l(A_l)$ を算出しプロセス待ち時間とする。

プロセス待ち時間を正しく計算するためには、 $M/G/K_l$ で系内人数 j に依存するサービス率 $\mu(j)$ をもつモデルを解析しなければならない。このモデルのサービス時間分布 (G) には、プロセスがアクティブ状態である時間の分布、すなわち待ち行列網内を移動する客の網内経過時間の分布が対応する。しかし、現在までのところ網内経過時間の分布に関してはあまり多くのことは知られていないので、本方法ではこの網内経過時間の分布を M でおきかえて計算を行う。正しいモデル $M/G/K_l$ における平均系内人数は、 $A_l = \lambda_l E_l(u_l)$ と計算され既知の情報となるので、同じ平均系内人数をもつ $M/M/K_l$ にモデルを置き換え待ち時間の計算を行う。

この方法による近似の精度については今後の検討に残されている。

5. QM-X の機能概要

QM-X は FORTRAN で記述されたソフトウェア・パッケージである。本章では、その入出力情報および処理の流れについて示す。

5.1 入力情報

QM-X が必要とする入力情報は次のごとくである。

(1) システム構成に関する情報

N : システム資源の総数

m_i ($i=1, 2, \dots, N$): 資源 i のサーバ数(無指定は $m_i = 1$)

M : サブシステムの総数

L : 閉鎖型サブシステムの総数

K_l ($l=1, 2, \dots, L$): サブシステム l のプロセス数

(2) ジョブ特性に関する情報

$r_{li} = (\rho_{1i}, \rho_{2i}, \dots, \rho_{N_i}), l \in \{\text{ジョブ特性を指定するサブシステムの番号}\}$: ジョブ特性ベクトル (ρ_{li} は、ジョブ当りの資源 i の総使用時間)

$r_{lij} = (\rho_{1ij}, \rho_{2ij}, \dots, \rho_{N_{ij}}), j=1, 2, \dots, J_l, l \in \{\text{ワークロードを指定するサブシステムの番号}\}$: ワークロード特性ベクトル (ρ_{lij} は、サブシステム l に属する j 番目のジョブ当りの資源 i の総使用時間)

$\varphi_{lij}, j=1, 2, \dots, J_l, l \in \{\text{ワークロードを指定するサブシステムの番号}\}$: サブシステム l に属する j 番目のジョブの相対発生率, $\sum_{j=1}^{J_l} \varphi_{lij} = 1$

(3) 負荷条件に関する情報

$\lambda_l, l \in \{\text{到着件数指定のサブシステム番号}\}$: 単位時

間当りのジョブの到着件数

$u_l, l \in \{\text{ジョブ空き時間指定のサブシステム番号}\}$:

サブシステム l に属するプロセスのアイドル時間

最大処理件数指定のサブシステムの場合にはその旨を指定する。負荷条件の指定方法とサブシステムの型の間には表1の整合性が保たなければならない。

5.2 出力情報

出力される性能評価レポートには、入力された情報、およびそれらに基づいて計算した結果得られる情報がともに表示される。さらに、全体像を概括的に把握するためのレポートから、システムの細部に至る詳細な情報を表示するレポートに至るまで、各種のレベルの情報が選択出力可能となっている。以下にはこのうち、計算結果として得られる情報について示す。

$a_i, i=1, 2, \dots, N$: 資源 i の使用率

$w_{li}, i=1, 2, \dots, N, l=1, 2, \dots, M$: サブシステム l に属する1ジョブ当りの資源 i についての待ち時間

ただし、 s_{li} をサブシステム l のジョブの1回当りの資源 i の待ち時間とすると、 $w_{li} = \theta_{li} s_{li}$ 。

$E_l, l=1, 2, \dots, M$: サブシステム l に属するジョブの処理時間 (資源使用時間+資源待ち時間)

$\lambda_l, l \in \{\text{ジョブ空き時間指定, または最大処理件数指定をしたサブシステムの番号}\}$: 単位時間当りのジョブ処理件数

$W_l, l \in \{\text{到着件数指定をしたサブシステム番号}\}$: プロセス待ち時間

$R_l, l \in \{\text{同上}\}$: トランザクション応答時間 ($E_l + W_l$)。 $A_l, l=1, 2, \dots, M$: サブシステム l の平均アクティブプロセス数 (マルチプログラミング多重度)

以上の項目は、必要に応じて、詳細情報として、サブシステムごと、さらにはワークロードごとの内訳が計算される。

5.3 処理の流れ

QM-X の処理は次の三つの部分からなっている。第1の部分はモデルを作成したり変更したりする部分であり、作成されたモデルはファイルに格納される。第2の部分は実行部分であり、利用者の指定する条件に従って計算を実行する。計算結果は予測ファイルに格納される。第3は編集出力部で、利用者の指定する編集方式に従って編集出力を行う。図2に処理の流れの概略を示す。

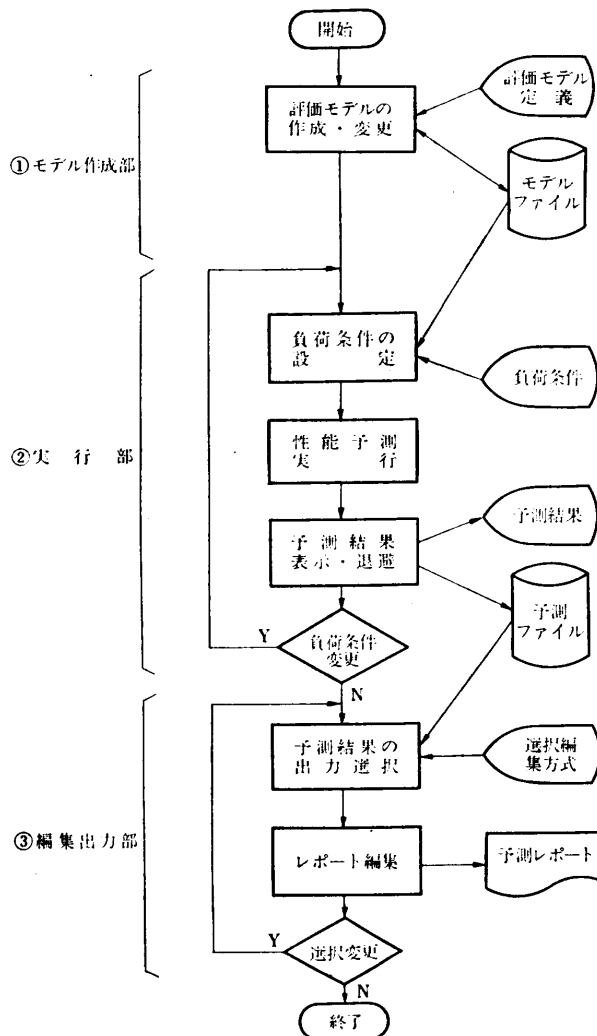


図2 QM-X の処理の流れ
Fig. 2 Processing flow of QM-X.

表 2 サブシステムの特徴
Table 2 Characteristics for subsystems.

サブシステム	(1) バッチ処理	(2) リアルタイム処理		(3) TSS
タイプ	閉鎖型	閉鎖型		開放型
プロセス数	$K_1=3$	$K_2=5$		—
ワークロード	—	照会処理 $\rho_{21}=0.7$	更新処理 $\rho_{22}=0.3$	—
負荷条件指定	50 件/時間 または 最大処理件数	λ 件/秒		0.1 コマンド/秒

表 3 ジョブ特性
Table 3 Job characteristics.

システム資源		リアルタイム・トランザクション		TSS コマンド
No.	名称	ρ_{41} (s)	照会処理 ρ_{21} (ms)	更新処理 ρ_{22} (ms)
1	CPU	20	200	400
2	DISK 1	10	120	120
3	DISK 2	10	150	210
4	DISK 3	20	—	—
5	DISK 4	10	—	—
6	DISK 5	—	210	210
7	DISK 6	—	—	500
8	DISK 7	—	—	120
計		70 /ジョブ	680 /TR.	1,440 /TR.
				900 /コマンド

6. QM-X の使用例

本章では簡単な使用例を示す。

6.1 モデル

バッチ処理, リアルタイム処理, TSS の 3 サブシステムが混在する多次元処理システムを例にとる. 前 2 者を閉鎖型, TSS のみを開放型とし, サブシステムの特徴を表 2 に示す. システムは CPU 1 台, DISK 装置 7 台から構成され, DISK 1 にはシステムファイル, DISK 2~6 はデータベースが格納され, DISK 7 は TSS 専用ファイルが格納されているものとする. 表 3 にジョブ特性を示す. リアルタイム処理に関しては, 照会処理, 更新処理の 2 種類のワークロードが設定されている.

6.2 評価例 1

トランザクションの到着件数 λ と応答時間の関係を調べたものを図 3 に示す. この例では, バッチジョブ到着件数を 50 件/時間, TSS コマンドの到着率を

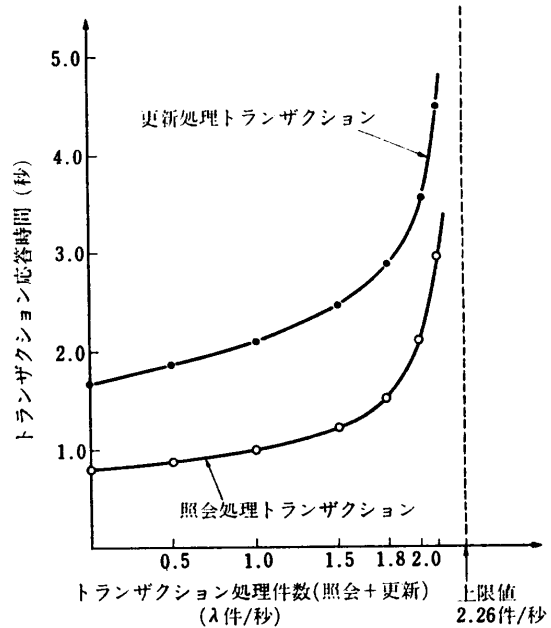


図 3 トランザクション処理件数と応答時間の関係
Fig. 3 The response time for transactions.

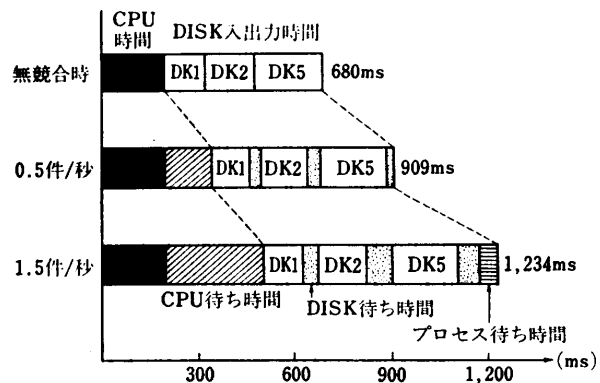


図 4 照会処理トランザクションの応答時間分析
Fig. 4 Analysis of transaction response time.

0.1 件/秒とした. さらに, 照会処理トランザクションを例にとり, 応答時間の内訳の分析を図 4 に示す. 本例では, CPU 待ち時間の増加が DISK 待ち時間の増加に比較して著しく, CPU がボトルネックになっていることがわかる. 同様に, CPU を高速なものに代えた場合や, DISK 装置を増設した場合の応答時間の改善程度について調べることができる.

6.3 評価例 2

バッチジョブのスループットと, トランザクション応答時間の関係を, トランザクション到着件数 λ をパラメータにして調べた結果を図 5 に示す. 本例では, バッチサブシステムの負荷条件の指定方法を, 先

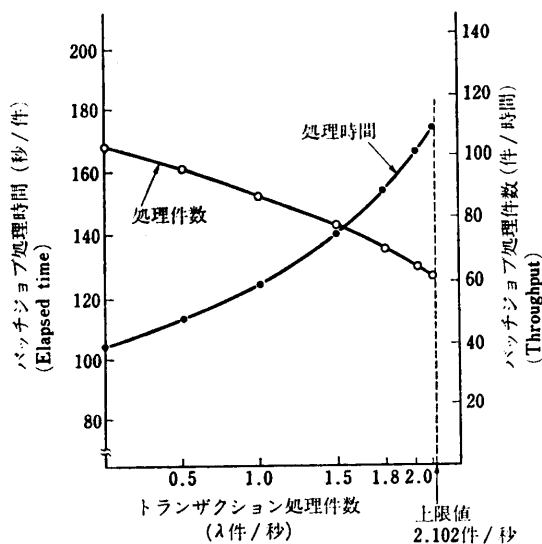


図5 バッチジョブの処理時間と処理件数

Fig. 5 Throughput and elapsed time of batch jobs.

の50件/時間に代えて、最大処理件数指定とする。この場合には、TSSおよびリアルタイムサブシステムに関しては指定された処理件数を処理した上で、なお余裕のあるシステム能力のすべてをバッチジョブの処理に向けたとしたときのバッチジョブの最大処理件数(スループット)を計算することになる。図5に示されるように、トランザクション処理件数 λ の増加に伴いシステムの余力が低下し、バッチジョブのスループットが低下する一方、1ジョブ当りの処理時間が増加してくる。本例のように、二つのサブシステム間の相互の干渉が簡単に評価できる点がQM-Xの特長の一つである。

7. まとめ

BCMP型待ち行列網モデルを基礎とする計算機システムの性能評価方法と、その実現であるソフトウェア・パッケージQM-Xについて述べた。

QM-Xは、1981年10月に第1バージョンをリリースして以来、現在までに社内の約80ユーザにリリースされ、手軽に利用でき、複雑なシステムの性能を的確に評価できる方法としてさまざまなシステムの性能評価に広く利用されてきている。利用状況に関しては、利用者へのアンケート調査に基づく結果が文献8)に示されている。

利用者から寄せられた意見に基づいて、QM-Xの今後の課題を整理すると次の二つに要約される。第1の課題は入出力情報に関連したものである。利用者か

らは把握しにくいOSのサービスルーチンやプログラムの動きに関する基礎データを内蔵し、モデルを自動的に作成する機能をもつこと、出力結果をグラフ表示したり複数の評価ケースを一覧表にまとめたりする編集機能を強化すること、また、安定に計算が実行できるパラメータの範囲や関係を明示すること、等である。第2の課題は評価機能の強化に関するもので、仮想記憶領域の制御機能の提供、優先権付サービス規律、通信系を含めた評価機能の提供、等である。第1の課題に関しては、OSの動作に関する基礎データを内蔵し、ジョブ特性を簡易に作成する機能や出力結果編集機能の強化の開発が現在行われている。第2の課題に関しては、モデルはすべて積形式解をもたない待ち行列網となり、これらに関する近似解法の研究が残されている。また、たたみこみ計算法に伴う指数部のあふれ現象を防ぐための効果的なスケールリング法の開発も今後の課題として残されている。

待ち行列網モデルによる評価の信頼性に関しては、1章に述べたように数多くの報告がなされているが、なお実システムとの比較検討によりその有効性と限界を明確にするための検証が残されている。この際には、入力情報を作成する際に発生する誤差と、モデルそのもののもつ誤差とをはっきりと区別して考察する必要があると思われる。このような観点から、QM-Xの評価例を整理し実測値との比較検討を行うことは意味深いと思われるが、紙幅の関係で別稿としたい。

謝辞 本研究の機会を与えていただき、途上数々の支援をいただいた、当社C&Cシステム研究所、三上徹所長代理、竹谷誠研究課長ならびに、情報処理営業支援本部、小沢五郎本部長代理、北浦隆課長、東北日本電気ソフトウェア(株)高橋晃社長に感謝の意を表します。また、QM-Xのプログラム開発に多大な協力をいただいた小林義昭氏、教育普及活動および設計支援をしていただいた守田節雄主任に感謝の意を表します。

参考文献

- 1) Baskett, F., Chandy, K. M., Muntz, R. R. and Palacios, J.: Open, Closed, and Mixed Networks with Different Class of Customers, *J. ACM*, Vol. 22, No. 2, pp. 248-260 (1975).
- 2) Reiser, M. and Kobayashi, H.: Queueing Networks with Multiple Closed Chains: Theory and Computational Algorithms, *IBM J. Res. Dev.*, Vol. 19, No. 3, pp. 283-294 (1975).
- 3) Reiser, M.: Modeling of Computer System

- with QNET 4, *IBM Syst. J.*, Vol. 15, No. 4, pp. 309-327 (1976).
- 4) Buzen, J. P. et al.: BEST/1—Design of a Tool for Computer System Capacity Planning, Proc. AFIPS NCC, pp. 447-455 (1978).
 - 5) 池原, 山田: 汎用ネットワーク形待ち行列解析プログラム QSEC について, 電子通信学会通信部門全国大会, p. 161 (1978).
 - 6) Bruell, S. C. and Balbo, G.: *Computational Algorithms for Closed Queueing Networks*, p. 127, North-Holland, Amsterdam (1980).
 - 7) Potier, D.: Performance Modeling Tools, Proc. International Symposium on Applied Mathematics and Information Science, Kyoto Univ., pp. 4.1-4.8 (1982).
 - 8) 納富, 北浦: 性能予測ツール QM-X とその応用, 情報処理学会, 計算機システムの制御と評価研究会, 17-2 (1982).
 - 9) 紀 一誠: 混合型待ち行列網の計算方法, 情報処理学会論文誌, Vol. 24, No. 3, pp. 326-334 (1983).
 - 10) 橋田 温, 川島幸之助: 待ち行列ネットワークモデルによる計算機システムの性能評価, 情報処理, Vol. 21, No. 3, pp. 743-750 (1980).
 - 11) Buzen, J. P.: A Queueing Network Models of MVS, *Comput. Surv.*, Vol. 10, No. 3, pp. 319-331 (1978).
 - 12) Lo, T. L.: Computer Capacity Planning Using Queueing Network Models, Proc. PERFORMANCE '80, pp. 145-152 (1980).
- (昭和 58 年 6 月 23 日受付)
(昭和 58 年 12 月 13 日採録)