システム性能評価モデルの構造的バリデーション手法について

村松　洋，伊達政広
（富士通　株式会社）

1. はじめに

待ち行列ネットワークモデルは，計算機システムの性能評価で広く用いられている。 モデリングによる性能評価作業は通常，以下のようなステップに大きく分類できる。

（1.a） モデルの構築及び，結果を得るステップ

（1.b） バリデーションと呼ばれる，モデルの妥当性を検証するステップ

（1.a）に対しては，Ｂｕｚｅｎ〔1〕，Ｂａｓｋｅｔｔ，Ｃａｎｄｙ〔2〕を始とする，クローズ型待ち行列ネットワークモデルの解法の著しい発展や，それを使用した計算用ツールの出現により，1970年代，後半以降大きな発展をとげている。 しかし，（1.b）については，モデリングにおいて非常に重要なステップであるにも，かかわらず，近年大きな進歩はない。

そこで，今回，我々は待ち行列ネットワークモデルを用いて，計算機システムの性能評価を行う場合に有効な，バリデーション手法の１つ（構造的バリデーション手法と呼ぶ。）を提案する。

2. 計算機システムモデルのバリデーションについて

モデルから得られた結果のバリデーション（検証）は検証されるものにより，以下のように分類される。

（2.a） モデルの構造の妥当性の検証

（2.b） モデルの基礎値の妥当性の検証

（2.c） モデルの計算方法特に近似の妥当性の検証

（2.c）に対する一般的な，アプローチは，複数の方法により，解を求め，結果を比較することにより行う。具体的にはシミュレーションにより求めた結果とアナリティクモデルにより求めた結果を比較するといったような方法である。 しかし，（2.a），（2.b）については，一般的な手法はいまだに確立されていない。

計算機システムにおけるモデリングの目的は，以下の２つに大別できる。

（2.d） 実在しないシステムの予測

（2.e） 実在するシステムを変更する場合の予測

（2.d）の目的のために作られたモデルのバリデーションは非常に難しいと，思われる。 本報告では，（2.e）についてのモデリング，すなわち実在するシステムのモデル化について，（2.a）モデルの構造，（2.b）モデルの基礎値を検証する方法を議論する。

近年，大型計算機システムは，ソフトウェア，ハードウェアともに益々複雑化する傾向にあり，システム全体を理解することは，非常に困難になっている。 すなわち，実システムに対する知識のみで，システムの振舞を表現するに，必要十分なモデルを，構築することは，もはや不可能になりつつある。 たとえば，ＴＳＳのモデルとして，最も手軽に使用できるものとして，“Ｓｔｒａｉｇｈｔ－ｆｏｒｗａｒｄ ｍｏｄｅｌ”〔3〕，と呼ばれるものがある。 このモデルによって，実システムの性能評価を行っても，ある程度の傾向の予測は可能であるが，得られる結果の精度は低い。 精度を上げるためには，もっと多くの要素をモデルに組み込む必用がある。 では，どこまでモデルを複雑化すれば良いのか。 モデルを複雑化し，独立なパラメタの数を増やすと結果を実測値に合せやすくなるが，モデルの妥当性が向上している訳ではない。

この問題に対する一つのアプローチとしては，“Ｏｐｅｒａｔｉｏｎａｌ Ａｎａｌｙｓｉｓ”〔4〕と呼ばれるものがある。 これは検証不可能な，モデルの細密化を考えるのではなく，検証可能な仮設とそれに基づいて，導出される諸関係式に着目することににより，明確な根拠を持った性能予測を行おうとするものである。

たしかに，この方法は実用的ではあるが，その構造からモデルに対する制約が厳しく，予測能力という点から，必ずしも十分とはいえない。 そこで，本報告は一般的な待ち行列ネットワークモデルにおいて，その構造をバリデ

ションするための１っの方法を提案する。

3. 待ち行列ネットワークモデルの線型性について

単純な，待ち行列ネットワークモデルは通常，図1.に示すような構成となる。これらのモデルは通常，ジョブとキューからなる。 キューは各々，サービス率とスケジューリング特性を持ち，ジョブは１っのキューから，他のキューへ，その遷移確率で移動する。 これは，我々は待ち行列ネットワークモデルの構造と呼ぶ。 又，あるキューに着目したとき，ジョブがｉを出発してから，再びキューｉに戻るまでの時間（Ｒｉ）を，キューｉからみたレスポンスタイムと定義する。 すなわち，キューｉのレスポンスタイムは，あるジョブがキューを出発して，再びキューｉに戻るまでに，通過した各キューでのサービス時間と待ち時間の総和であり，そのジョブがキュー ｊに滞留した時間をＯｊとすると，レスポンスタイム（Ｒｉ）は以下の式（1）のようになる。

$$Ri = \Sigma Oj \qquad \cdots\cdots (1)$$

又，Ｏｊはキュー ｊでの滞留時間の平均値Ｐｊ（サービス時間と待ち時間）とキュー ｊへの訪問回数Ｎｊの積である。今，Ｒｉはキューｉを出発してキューｉに戻るまでにＮ１からＮｍのキューを経由するとすると，Ｒｉは以下のような，線型１次式で，表現できる。

$$Ri\ (N1, N2, N3, \cdots, Nj, \cdots Nm) = \sum_{j=i}^{m} Nj \cdot Pj \qquad \cdots\cdots (2)$$

式（2）が成り立つ厳密な，条件は判っていないが，単純なモデルで成立することは自明と思われる。 この仮定をもとに，モデルの妥当性を検証するための手順を以下にしめす。

（3.a） 計算機システムを表現するために，必用と思われるキューを設定し，初期モデルを作る。

（3.b） 実測により，ＲｉとＮｊを得，Ｒｉを従属変数Ｎｊを独立変数として回帰分析により，ＲｉとＮｊの相関を求める。

（3.c） ＲｉとＮｊが１次近似でよく表現されており，かつＰｊが妥当なら，キューは妥当とする。 妥当でないなら，モデルを再定義する。

4. CASE STUDY

我々は，前述の方法により，ＯＳⅣ／Ｆ４ ＴＳＳ〔5〕の性能評価を行った。ＯＳⅣ／Ｆ４ ＴＳＳは超大型システム用の汎用ＴＳＳシステムであり，スケジューリングアルゴリズムを始，その内部構造はきわめて，複雑であるが，我々はとりあえずの出発点として，図2.に示すＴＳＳモデルを設定した。

1） ＯＳⅣ／Ｆ４の大規模ＴＳＳユーザの実稼動システム４センターについて，ＧＴＦ〔6〕と呼ばれるイベントトレーサにより，端末入出力，ディスパッチング，ＤＡＳＤ入出力，ページフォルト などの情報を収集した。

次に，そのトレースデータをＴＳＳ１コマンド単位に，編集し，コマンド単位に，ＣＰＵ時間（Ｓｃｐｕ），ＤＡＳＤ Ｉ／Ｏ回数（Ｎｉ／ｏ），ユーザ領域でのページフォルト回数（Ｎｐａｇｅ１），システム領域でのページフォルト回数（Ｎｐａｇｅ２），スワップ回数（Ｎｓｗａｐ），レスポンスタイム（Ｒ）をもとめた。

2） 1）で得られた数千のコマンドについて，式（3）の線型１次式をあてはめ，Ｃ０，Ｃ１，Ｃ２，Ｃ３，Ｃ４，Ｃ５を重回帰分析により，算出した。

$$R(Scpu, Ni/o, Npage1, Npage2, Nswap) = C0 + C1 \cdot Scpu + C2 \cdot Ni/o + C3 \cdot Npage1 + C4 \cdot Npage2 + C5 \cdot Nswap \qquad \cdots\cdots (3)$$

ＣＰＵのキュ に対しては訪問回数（Ｎｃｐｕ）の代りにＣＰＵ時間を収集しており，Ｃ１・ＳｃｐｕはＣＰＵキュ におけるサービス時間と待ち時間の和を表す。

表1.に，Ｃ０～Ｃ５の計算結果と，通常の環境での期待値を示す。 表1.の結果から，以下のことがいえる。

（4.a） ページフォルトが多発していないセンターでは我々の設定した，単純なモデルでも比較的，良くレスポンスタイムを表現している。

（4.b） スワップキューは細密な予測を得る必要がなければ削除できる。

（4.c） ページフォルトが多発しているセンタではモデ

ルはシステムを十分に表現していない。（C3，C4の値はページフォルトを解決するための処理，すなわち，アクセスしたページが主記憶にないために，DASD上から必要なページを読み込むための時間が大き過ぎる。）

最初に設定したモデルに対して，ページング部分をより細密化する必要があることをしめしている。

具体的には図3.に示すような，モデルが考えられる。

5. 結論

計算機システムの性能評価モデルのバリデーションは，従来は，たとえば，CPU使用率などについて，実測と比較することによって，モデル全体の妥当性を評価するような方法で行われていた。 我々の提案した構造的バリデーション手法はレスポンスタイム（Ri）と各キューの滞留時間（Oj）との1次近似の成立範囲など，十分には検証されていないものもあるが，以下の点で従来の方法よりも

(1) モデルのどの部分をより細密にすべきか

(2) モデルを単純化するためには省略すべきものは何か

(3) 本来，そのモデルでレスポンスタイムがどの程度，表現できているのか

という，問題に対して具体的な指針を得ることができ，きわめて有効な方法とおもわれる。

References

〔1〕J. P. Buzen, "Computational Algorithms for Closed Queueing Network with Exponential Server," C. ACM, Vol. 16, N. 9, 1973, pp. 527－531

〔2〕F. Baskett and K. M. Candy "Open, Close, and Mixed Networks of Queues with Different Classes of Customer," J. ACM, Vol. 22, N. 2, 1975, pp. 248－260

〔3〕J. W. Boyse and D. R. Warn, "A Straightforward Model for Computer Performance Prediction," Computing Surveys, Vol. 7, N. 2, 1975, pp. 73－93

〔4〕P. J. Denning and P. J. Buzen, "Operational Analysis of Queueing Network Models1," Computing Surveys, Vol. 10, N. 3, 1978, pp. 225－261

〔5〕OSⅣ／F4 解説書，64SG－1002，富士通，東京

〔6〕OSⅣ／F4 サービスエイド使用手引書 64SP－1252，富士通，東京

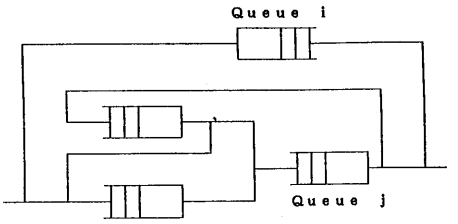

図1.　一般的な待ち行列ネットワークモデルの例

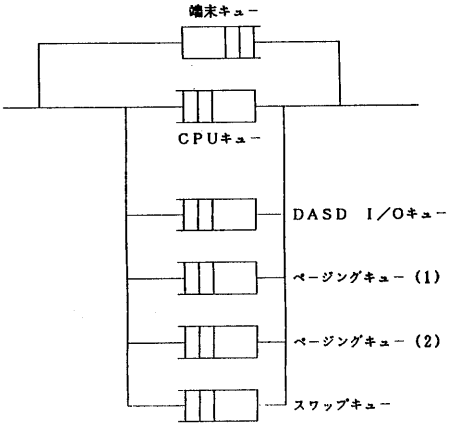

図2.　OSⅣ/F4　TSSの初期モデル

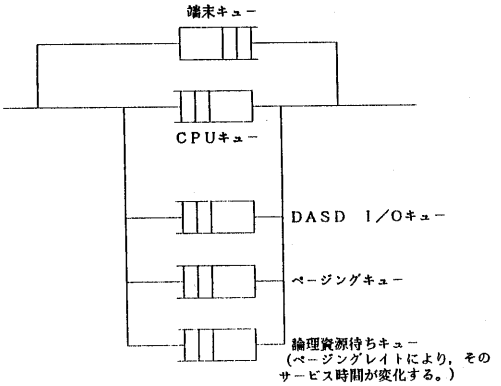

図3.　OSⅣ/F4　TSSのバリデーション後のモデル

表1.　各センタでの評価値

回帰係数	モデルでの期待値	Aセンタ	Bセンタ	Cセンタ	Dセンタ
C0 (定数)	0	0.26	0.21	−1.05	−0.14
C1 (CPUのサービス時間と待ち時間)	$\frac{1}{1-\rho *}$	1.10	1.57	3.67	3.82
C2 (DASD I/O時間)	20~50 m.s.	46	37	50	21
C3 (ページング I/O時間)	15~50 m.s.	93	39	121	86
C4 (ページング I/O時間)	15~50 m.s.	157	18	63	71
実測により得られた ρ * ρはTSSユーザのCPU使用率		1.16	1.19	3.70	1.90