B-030

待ち行列理論による抽象化を用いたモデル検査手法の検討
Study of Model Checking by Using Abstraction Based on Queueing Theory

大林 浩気† 長野岳彦† 茂岡 知彦†

Hiroki Ohbayashi Takehiko Nagano Tomohiko Shigeoka

1. はじめに

近年，組込みソフトウェアの複雑化・大規模化が進み，開発下流工程での手戻りコストが増大している．手戻りの原因の一つとして，性能目標未達などの問題がある．このような背景から，上流工程における性能検証技術の重要性が高まっている．

上流工程で用いられる検証技術としてはモデル検査が知られている．しかし，応答時間など時間に関する性能検証に対してモデル検査を適用する場合，時間変数の組合せ増加によって状態数爆発が発生しやすいという問題がある．例えば，複数コンポーネントの処理の組合せでは，コンポーネントに対する処理要求の到着時間間隔に応じて処理開始までの待ち時間が発生し，その長さは様々な値を取りうる．その結果時間変数の組合せが増加し，状態数爆発が起こる．

この問題の解決方法として，我々は検証モデルを待ち行列モデルにより抽象化することによって状態数爆発を回避する手法を提案している．具体的には，待ち行列理論により算出した平均待ち時間で待ち時間の値を一通りに集約することによって，モデル検査実行に必要な状態数を削減する．本稿では，複数コンポーネントの処理の組合せを対象とし，性能検証用 Promela モデルの作成方法の概略と，待ち行列モデルによる抽象化方法について述べる．また，提案手法の評価実験についても述べる．

2. 提案手法

本節では提案手法について述べる．適用対象として，図 1 のような UML ダイアグラムで表現されるシステムの性能検証を考える．Caller はユーザーのリクエスト到着によって動作を開始し，Server に処理を要求する．Server は Caller から処理要求を受信すると処理を開始し，処理が完了すると Caller に処理結果を返却する．ただし，Server が処理中であるときに処理要求を受信した場合，処理要求は FIFO バッファにキューイングされ，処理開始を待つ．

2.1 性能検証用 Promela モデル

上記システムからモデル検査のための時間概念を含む検証モデルを作成する方法を述べる．モデル検査にはモデル検査ツール SPIN[1]に離散時間の機能を拡張した DT-SPIN[2]を使用する．DT-SPIN では整数の値を持つタイマーと呼ばれる要素によりモデルに時間の概念を持たせることができる．DT-SPIN の Promela モデルでは，引数で指定した値だけタイマーが減少するまで実行をブロックする delay マクロを挿入することで，その箇所における時間の経過を表現することができる．

Promela モデル作成の手順を以下に示す．

i. 各 Server に対応する，プロセス，処理要求をキューイングするバッファ付チャネルおよび処理結果を返却するチャネルを定義する．プロセス内にはループを作成し，ループの開始点には処理要求チャネルの受信命令，終了点には処理結果返却チャネルの送信命令と処理時間だけ時間を経過させる delay マクロを記述する．

ii. Caller に対応するプロセスを定義する．プロセスには各 Server の処理要求チャネルの送信命令と処理結果返却チャネルの受信命令を，シーケンス図のライフラインの通りに時系列順に記述する．

iii. Caller に対するユーザーのリクエスト到着を表現するための Arrival プロセスを定義する．Arrival プロセス内にはループを作成し，ループ内には Caller プロセスの起動命令と，リクエストの到着時間間隔だけ時間を経過させる delay マクロを記述する．

iv. init プロセスを定義する．init プロセス内には Arrival プロセスと各 Server プロセスの起動命令を記述する．

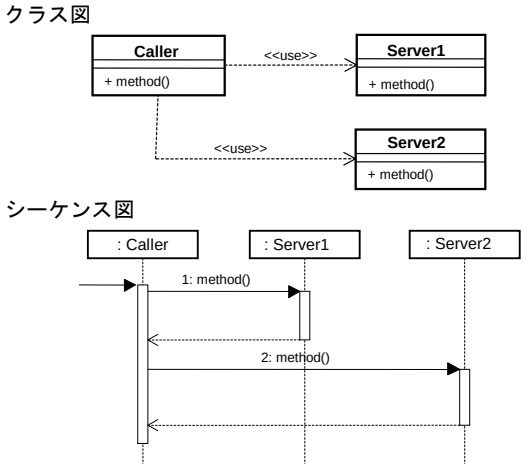

図 1 適用対象の UML ダイアグラム

2.2 待ち行列モデルによる抽象化

前項で述べた検証モデルを待ち行列モデルにより抽象化する方法について述べる．例えば，図 1 の UML ダイアグラムで表されるシステムは，図 2 のような待ち行列ネットワークとみなせる．提案手法ではこの待ち行列ネットワークを解析することにより平均待ち時間を算出し，検証モデルに組込む．以下その方法について述べる．

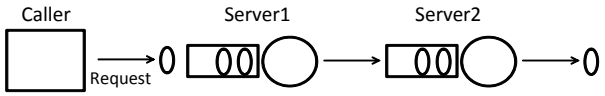

図 2 待ち行列ネットワーク

2.2.1 平均待ち時間の算出アルゴリズム

まず，待ち行列モデルによる抽象化適用時に必要となる平均待ち時間の算出方法について述べる．到着時間間隔，処理時間のどちらの分布も指数分布である M/M/m 型待ち行列モデルについてはその平均待ち時間 $W_q(M/M/m)$ の公式が知られている．ところが，指数分布に従う確率変数の値域は [0,∞)であるのに対し，コンポーネントの処理時間は最良値と最悪値が存在するのが自然である．したがって本稿では，処理時間の分布を指数分布と仮定するのは不適切であり，M/M/m 型待ち行列モデルによる解析を用いることはできない．

以上のことから，本稿では GI/G/m 型待ち行列モデルを用いることとする．GI/G/m 型待ち行列モデルによる解析においては平均待ち時間の近似値しか得ることができないが，到着時間間隔や処理時間について任意の分布を仮定することができるという利点がある．平均待ち時間の近似方法については様々に研究がされているが[3]，本稿では最も基本的である以下の式を用いる．

$$W_q(C_a, C_s, m) = W_q(M/M/m)(C_a^2 + C_s^2)/2 \qquad (1)$$

ただし，C_a は到着時間間隔の分布の変動係数であり，C_s は処理時間分布の変動係数である．

GI/G/m 型待ち行列モデルにおいては待ち行列ネットワークの解析理論が確立していない．そのため複数の待ち行列を含むモデルに対して抽象化を適用する場合に問題となる．本稿の対象では待ち行列ネットワークの各ノードにおいて分岐や外部からの流入および外部への流出が無いことを利用し，各ノードの到着過程を前ノードの出発過程と同一とすることでこの問題を解決する．ここで，平衡状態を考えることにより全ノードの平均出発時間間隔はユーザーからのリクエストの平均到着時間間隔と等しいと仮定して問題ない．また，出発時間間隔の変動係数 C_d は Marshall の公式と呼ばれる近似式を用いて以下のように算出できる[4].

$$C_d = 1 + (1-\rho^2)(C_a^2 - 1) + (\rho^2 C_s^2 - 1)\rho^2/\sqrt{m} \qquad (2)$$

図 2 の GI/G/m 型待ち行列ネットワークに対する具体的な平均待ち時間の算出アルゴリズムを以下に示す．ただし，Server1，Server2 の窓口数，処理時間の平均と分散，および，Server1 の到着時間間隔の平均と分散は最初に与えられているとする．

i. 与えられている Server1 の到着時間間隔の平均と分散および処理時間の平均と分散から，それぞれの変動係数を求める．
ii. 与えられている Server1 の窓口数，処理時間の平均，到着時間間隔の平均と，i で求めた変動係数を用いて式(1)から Server1 における平均待ち時間を求める．
iii. 同様に式(2)から Server1 の出発時間間隔の変動係数を求める．
iv. Server2 の到着時間間隔について，平均を Server1 の到着時間間隔の平均と同じとし，変動係数を iii で求めた値とする．
v. 与えられている Server2 の処理時間の平均と分散から，その変動係数を求める．
vi. 与えられている Server1 の窓口数，処理時間の平均と iv, v で求めた到着時間間隔の平均，変動係数を用いて式(1)から Server2 における平均待ち時間を求める．

上記手順を繰り返すことにより，分岐，途中の流入・流出のない任意の長さの GI/G/m 型待ち行列ネットワークに対して，各ノードでの平均待ち時間の近似値を算出することが可能である．

2.2.2 検証モデルへの組込み

次に，上記算出方法で得られた平均待ち時間を検証モデルに組込む方法について述べる．平均待ち時間の組込みは Server プロセス単位で任意に選択が可能である．平均待ち時間の組込みを実施する Server プロセスに対して，Promela モデルを以下のように変更する．

i. Server プロセス内の delay マクロを削除する．
ii. Caller プロセス内の処理結果返却チャネルの受信命令後に，得られた平均待ち時間と処理時間をタイマーの値から減算する命令を挿入する．

これにより実行権が Server プロセス内でブロックされず，処理要求チャネルのキューにおける待ちが発生しなくなり，待ち時間の長さが一通りに集約される．

3. 評価実験

提案手法の有効性を調べるため，評価実験を行った．実験には ATM の残高確認を題材とした検証モデルを用いた．表 1 はモデル検査の全探索完了時の状態数を，待ち行列モデルによる抽象化を実施しない場合と実施する場合で比較したものである．抽象化によって状態数がおよそ 20%に削減されており，提案手法の状態数削減効果が確認できる．

表 1　全探索完了時の状態数の比較

	抽象化なし	抽象化あり
状態数	39049394	8297850
比率	1.00	0.21

4. まとめと今後の課題

本稿では，モデル検査による性能検証における状態数の削減を目的として，検証モデルを待ち行列モデルにより抽象化する手法を提案し，複数コンポーネントの処理の組合せを対象として具体的方法を述べた．また，評価実験により提案手法の有効性を確認した．

本稿で述べた抽象化方法は待ち行列ネットワークに分岐，途中の流入・流出のない場合のみ適用可能であり，平均待ち時間の算出アルゴリズム等の改善により適用可能な範囲を広げることが必要である．また，抽象化によって生じるモデルの性質の変化や検証精度の低下の影響についてはまだ議論が不十分である．これらの解決が今後の課題となる．

参考文献

[1] Spin - Formal Verification, http://spinroot.com/.
[2] Discrete-time Promela and Spin, http://www.win.tue.nl/~dragan/DTSpin/.
[3] T. Kimura, "A Two-Moment Approximation for the Mean Waiting Time in the GI/G/s Queue," Management Science, 32 (1986), 751-763.
[4] W.Whitt,"Departures from a Queue with Many Busy Servers," Math. Oper. Res., 9 (1984), 534-544.

†株式会社日立製作所横浜研究所　Yokohama Research Laboratory, Hitachi Ltd.