

柿元 満
M. Kakimoto

飯田 晴彦
H. Iida

藤原 睦
M. Fujihara

計算機システムの簡単で正確な性能評価ツールが、開発期間短縮や品質向上を旨とするシステムエンジニア (SE) やソフトウェアエンジニアから要求されている。従来から、性能評価ツールとして、待ち行列網理論をベースにしたモデルによるシミュレーションが利用されているが、モデルの作成が必ずしも容易ではなかった。ここでは、二つのシミュレーションツール“雲水”と“e ツール (仮称)”を紹介する。システム計画段階での性能予測用の“雲水”は、システム仕様からシミュレーションモデルを生成する。システム設計段階での性能検証用の“e ツール”は、その“OS モデル”機構により詳細なモデルの構築を支援する。どちらのツールもモデル作成を支援することによって、性能評価のためのシミュレーションの実用性をさらに高めている。

Performance prediction is an indispensable part of information systems development. A widely accepted modeling framework for this purpose is queueing networks, which are evaluated by discrete event simulation. The bottleneck in this approach is that the description of information systems in terms of queueing networks is sometimes found to be difficult.

This paper presents two tools which assist systems engineers in building queueing network models of information systems. They are (1) “UNSMUI”, which generates simulation models from computer system configuration specifications; and (2) “e-tool”, which automatically fills in the model details based on a built-in knowledge of the operating system.

1 まえがき

顧客の要求を満足させるために、SE はさまざまなハードウェア機器やソフトウェアを組み合わせてシステムを構築しなければならない。このとき機能的な仕様を満たすのは比較的容易であるが、ハードウェアのスペックやオペレーティングシステム (OS)、アプリケーションの動作特性を考慮して性能を定量的に見積もるのは容易ではない。安全を図るためあえて冗長なシステムを設計せざるをえない場合もある。これを解決するため設計段階のシステムを待ち行列網でモデル化し、シミュレーションによる性能予測が行われている。

待ち行列網では、システムをサーバ (資源に相当し、窓口ともいう。) とジョブ (資源利用者のことで、客ともいう。) から構成されるとみなす。ハードウェア機器や OS の提供する資源をサーバ、ユーザの行う処理の流れをジョブに対応させれば、おおまかには計算機システムのモデルが待ち行列網として記述できる。例えば、ディスク装置をサーバ、I/O 処理をジョブとみなせば複数の I/O 処理がディスクで競合する様子を表現できるし、CPU をサーバ、その上で動くプロセスをジョブとみなせば、マルチタスクのモデルが待ち行列で表現できる。このようなモデルを用いてシミュレーションを行った結果、あ

るサーバの待ち行列が長い、つまりサービス待ちのジョブが多いことがわかれば、そのサーバにクロックの速い CPU を用いて処理能力をあげたり、データを複数のディスクに分割して負荷を分散させるなどして対処することができる。

しかしながら、SE がシミュレーションプログラムを作るのは次のような問題のために必ずしも容易ではない。

- (1) モデルに設定すべきデータ量が多い。
- (2) 待ち行列が必ずしも計算機の動作記述に適していないので、待ち行列網モデルの表現に慣れる必要がある。
- (3) 計算機の資源管理アルゴリズム (特に OS) が複雑である。

ここでは SE をユーザとして想定した二つのツールを紹介する。これらのツールはそれぞれ上述の問題に対して異なったアプローチを取っている。

計算機応用システムの計画段階ではシステム構成などを決定するが、詳細なモデルで性能予測を行うよりも、粗いモデルで性能の概略を迅速に知りたい。“雲水”は、上述の(1)と(2)を解決する。システム提案時に、グラフィカルユーザインタフェース (GUI) を用いて設定したシステム構成案からシミュレーションモデルを自動生成し、提案システムの性能を簡単に予測することができるようになる。近年分散化が進んでい

る発電プラント監視システムや電力系統監視システムへこのツールの適用を試みている。

一方、計算機応用システムのシステム設計が進むと、計算機内部の動作を正確にシミュレートして、精度の高い性能見積もりが必要になるが、この場合、(2)、(3)の問題が重要になる。“e ツール”と“OS モデル”は部品ライブラリを提供することによってこれに対応する。待ち行列モデル部品を組み合わせることによって一からモデルを構築する必要がなくなる。特に、オブジェクト・モデルの継承の概念を利用して、多様な計算機システムに対応できる。このツールは納入段階での性能見積もりが困難なトランザクション処理システムに具体的に適用を試みている。

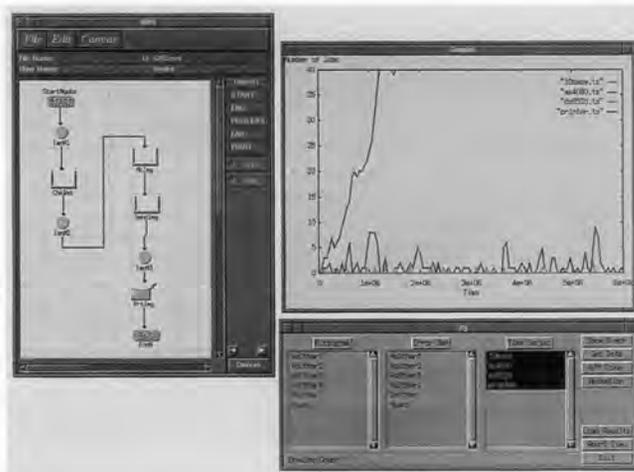


図2. “雲水”のGUI ハードウェア構成およびデータフローによってシステム仕様を与える。
GUI of “UNSUI”

る。ここではシステム仕様をハードウェア構成とデータフローによって表現する(図2)。また、データフローの各ノード(データ処理の単位)をハードウェア構成のどの要素に割り当てるか決めることもできる。ノードおよび要素に関する詳細な情報(属性)はテキスト形式で入力する。

2.2 シミュレーションモデル自動生成

ユーザが入力したシステム仕様からシミュレーションモデル(待ち行列網モデル)を自動的に生成する。シミュレーションモデルはシステム仕様のデータフローに基づいて作成される。データフロー中の各ノードが対応する待ち行列網モデルに順次変換される。変換後にモデルがどのような形になるかは、そのノードの種類や属性値によって決定される。また、モデルに必要な定量的な情報(パラメータ)も属性値から算出される。パラメータも含めたモデルへの変換パターンはユーザが設定することも可能であるが、通常はツール付属のものを使えばよい。

生成された待ち行列網モデルは使用されるシミュレータが読み込める形で書かれている。ここで用いられる待ち行列網シミュレータは、より多くのシステムの挙動を表現できるように、記述要素を細分化したものである。

ハードウェア、タスク、データなどのシステムリソース全般をエージェントと呼ばれる“物”で表す。エージェントはステーションと呼ばれる“場”を順次訪問する。このエージェントの訪問によってシステムリソースの状態遷移を表す。例えば、タスクを表すエージェントは、実行不可状態、実行待ち状態、実行中の状態を表すステーションを順次訪問する。

リソース間の同期、リソースの確保、サービスの開始などはジャンクションと呼ばれるステーション間の関係によって表現する。また、サービス授受の状態はコネクションと呼ばれる、エージェント間の結合関係によって表現する。図3は、

2 システム性能評価ツール“雲水”

このツールは次の三つの部分から成る(図1)。

- (1) システム仕様作成
- (2) シミュレーションモデル自動生成
- (3) 評価および結果提示

ユーザがシステム仕様の作成とシミュレーション実行を指示するだけで、対象システムの性能を評価し、性能を低下させる要因(ボトルネック)を検出することができる。また、提示された評価結果を基に、システム仕様を修正して再び評価することも可能である。

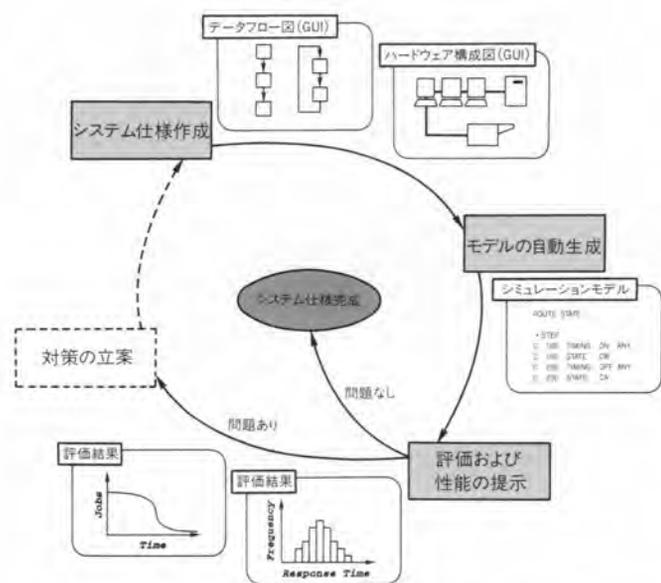


図1. “雲水”の構成 仕様作成部、モデル生成部、評価部の三つから成る。
Configuration of “UNSUI”

2.1 システム仕様作成

ユーザによるシステム仕様作成を GUI によってサポートす

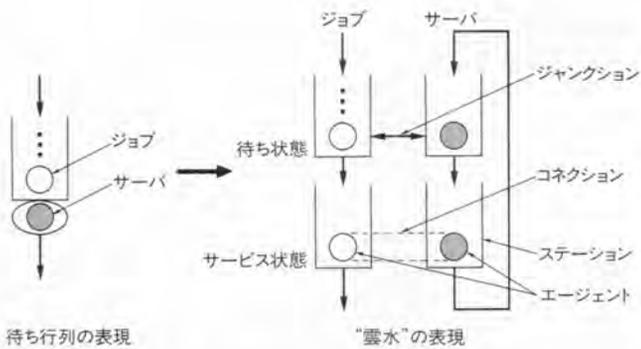


図3. シミュレーションモデル 細分化した待ち行列の機能を組み合わせることにより複雑なモデルを表現する。

Simulation model

通常の待ち行列がどのように表現されるかを示している。

このように、待ち行列の機能を細分化して個別に提供することにより、通常の待ち行列の枠組みには収まらない複雑なモデルもこれらの機能を組み合わせて構築することが可能となった。

2.3 評価および結果提示

生成されたシミュレーションモデルを待ち行列網シミュレータにより評価する。シミュレーション結果として、応答時間の分布、応答時間の時間変化、滞留数の時間変化のグラフを得ることができる。応答時間に関する結果はシステムの性能の把握に、滞留数に関する結果はボトルネックの検出に用いられる。

ユーザはシミュレーション結果から、システムが性能的な要求を満足させるか否かを判断する。性能的な問題がある場合は、ボトルネックがどこであるかなどを参考にして、性能改善のための対策を立案する。立案された対策をシステム仕様に反映すべく、GUIを用いてデータフローを修正し、再び評価を行う。シミュレーション結果が性能的な要求を満足させるまで、この過程が繰り返される。

3 “e ツール” と “OS モデル”

計算機システムは複数のハードウェア、ソフトウェアのモジュールが階層的に組み合わさって構成されている。このため、計算機の動作全体を一人のSEが見通して、待ち行列のモデルとして書き下すのは難しい。現実的なアプローチはモデルを個々のサブモデルに分割し、それぞれの部分の設計者がモデルを作成する。こうすればエンドユーザとなるSEはこれらのサブモデルを組み合わせるだけでシミュレーションが可能になり、サブモデル内の動作の詳細には煩わされる必要がない。

この際、技術的に重要なことは、適切なサブモデルへの分割とサブモデル間インタフェースの明確な枠組である。サブ

モデルへの切分けは現実の計算機システムの階層性やモジュール性に即して行うのが自然である。また、こうしたモデルの分割を表現するにはオブジェクト指向の考えかたが適している。

“e ツール”では、計算機システムをハードウェア、OS、アプリケーションの階層に分け、さらに必要に応じて各階層をより細かなサブモデルに分けた枠組を作り、それぞれをC++のクラスとして実現している。これを“OS モデル”と呼んでいる。このOS モデルはC++を用いて離散事象シミュレーションを行うためのクラスライブラリである“e ツール”を用いて実現されている。そこで、以下ではまず“e ツール”について、続いて“OS モデル”の概要、およびその考えかたを実際に適用したUNIX^(注1)モデルを説明する。

3.1 “e ツール”

“e ツール”は次のような特長をもつ(図4)。

- (1) プロセス指向 “e ツール”ではプロセス指向の考えを待ち行列網記述に採り入れた。その特長はルートをプログラム言語の処理フローとして(簡単にいうとif then やforを用いて)記述できることである。これにより、条件判断やループを含む複雑なルートの記述が容易になる。
- (2) 待ち行列以外のモデル化手法 計算機内には通信な

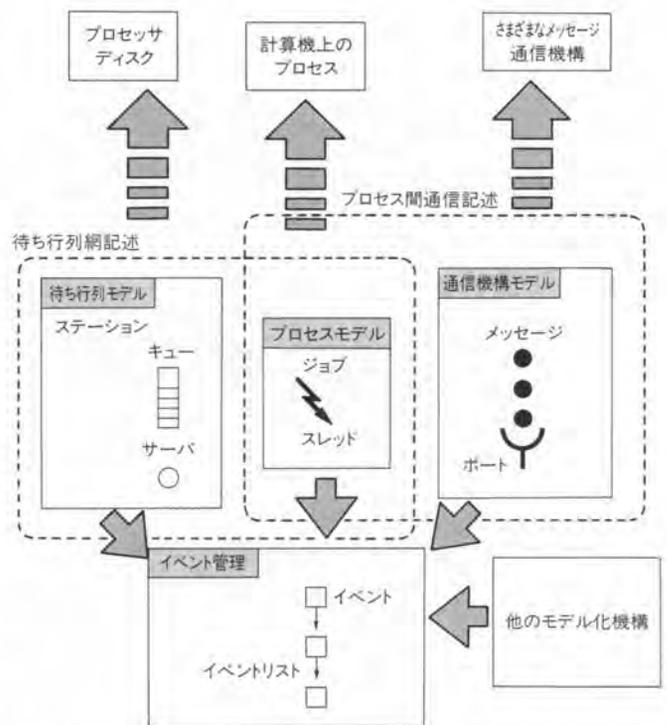


図4. “e ツール”の構成 待ち行列、プロセス間通信のモデルをもつ。継承によりモデルの拡張を図ることができる。

Configuration of “e-tool”

(注1) UNIXは、X/Openカンパニーリミテッドがライセンスしている米国ならびに他の国における登録商標。

ど待ち行列だけのモデル化が難しい部分もある。このため“e ツール”ではプロセス間の通信に基づいたモデルも提供している。また、ユーザが独自のモデルを作れるように、シミュレーションのイベント管理を直接操作できるようにしている。

3.2 オブジェクト指向の“OS モデル”の枠組

“OS モデル”は、次のサブモデルから構成される(図5)。

- (1) メモリ管理部モデル
- (2) プロセス(スレッド)管理部モデル
- (3) ファイル(バッファ管理, データ配置管理)管理部モデル
- (4) プロセス間通信部

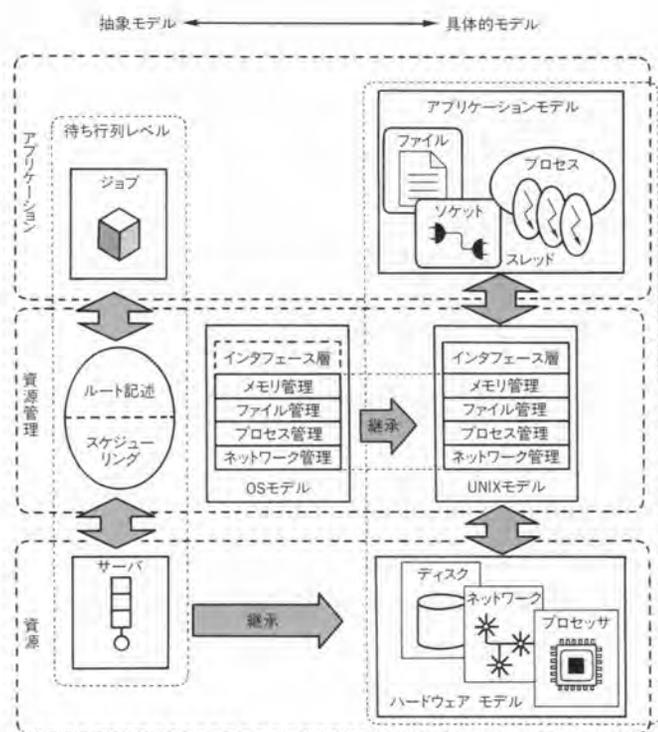


図5. “OS モデル”の構成 OS モデルは四つのサブモデルから構成される。UNIX モデルは UNIX の資源管理機構を組み込んでいる。

Configuration of operating system model

各サブモデルはそれぞれメモリ管理, スケジューリング, ディスク上のデータ配置, 通信のアルゴリズムを記述する部分で, それぞれC++の一つのクラスとして定義されている。

例えば, データの読み込みを例にとってサブモデルがどのように動作するか見てみる。まず主記憶上のバッファに所望のデータがあるか(バッファ管理)調べられる。ない場合は, データを格納するためのメモリを確保し(メモリ管理), ディスク上でのデータの格納場所を求め(データ配置管理)ディスクからの読み込み動作が行われる。このような複数資源にまたがる一連の処理の流れは多くのOSで共通に見いだせるので, サ

ブモデル相互の基本的な関係として組み込んである。

一方, サブモデル内部の動作は個々のOSによって異なる場合が多い。OSモデルではオブジェクト指向の継承の機構を利用してこれに対応する。それぞれサブモデルを定義したクラスの子クラスのなかで, 個々のOS固有の動作を記述する。これにより基本的な枠組は踏まえたうえで, モデルに多様性をもたせることができる。

また, OSモデルは外側に実際のOSと似たインタフェースをもっている。これによりアプリケーションの動作モデルを一般のプログラムと同じように記述できる。実際にあるOSでのプログラミングに親しんだユーザにとっては大きな利点となる。

3.3 UNIX モデル

UNIXモデルはUNIX独自の資源管理アルゴリズムと, その実行に必要なプロセッサ使用時間がパラメータとして組み込まれている。

UNIXモデルのインタフェースではUNIXのシステムリソース(例えば, プロセス, スレッド, ファイル, ソケットなど)を表現するオブジェクトがC++のクラスとして定義されている。アプリケーションの動作はこれらのオブジェクトに対する操作として記述できるので, UNIX上でのプログラミングに親しんだユーザであれば, シミュレーションプログラムの記述も容易である。

4 あとがき

計算機システムのシミュレーションモデルの作成を容易にする二つの性能評価ツール“雲水”と“e ツール”を紹介した。これらのツールを適用してシステム性能を把握することにより, それ以降の開発工程で性能問題が発生することが防止され, それに伴ってシステム仕様の変更による工程の後戻りが減り, 全体の開発期間を短縮する効果が期待される。



柿元 満 Mitsuru Kakimoto, D.Sc.

1987年入社。計算機システム性能評価技術の研究・開発に従事。現在, 研究開発センター 情報・通信システム研究所主務, 理博。

Communication & Information Systems Research Labs.



飯田 晴彦 Haruhiko Iida

1991年入社。計算機システム性能評価技術の研究・開発に従事。現在, 研究開発センター システム・ソフトウェア生産技術研究所。

Systems & Software Engineering Lab.



藤原 睦 Mutsumi Fujihara

1981年入社。離散事象シミュレーション技術の研究・開発に従事。現在, 研究開発センター システム・ソフトウェア生産技術研究所主査。

Systems & Software Engineering Lab.