可変構造マシンを用いた確率モデルシミュレーションシステム

山本 欧 (東京電機大学工学部), 柴田 裕一郎, 天野 英晴, (慶應義塾大学理工学部)

あらまし
並列システムの解析に用いられる数学モデルの中で，離散時間マルコフチェーン (DTMC) と待ち行列は，解析を専門としない並列システムの設計者にも習得が容易な確率モデルである．しかし，複雑なシステムをモデル化した場合や，発生確率の極めて小さい事象を扱う場合には解析に多大な時間を要する．そこで本論文では，乱数を用いたDTMC／待ち行列のシミュレーションを，可変構造マシン上でハードウェアによって高速に実行するシステムを提案する．このシステムでは，モデル記述言語Taicoで解析対象を記述し，それをトランスレータによってハードウェア記述言語に変換した後，可変構造マシン上に実現しシミュレーションを実行する．評価の結果，高速ワークステーションに比べて数百倍の速度でシミュレーションが可能であることが判明した．
キーワード：マルコフチェーン，待ち行列，可変構造マシン

A Reconfigurable Stochastic Model Simulation System

O Yamamoto (Tokyo Denki University), Yuichiro Shibata, Hideharu Amano (Keio University)

Abstract
Markov chain and queueing model are convenient tools to analyze parallel systems for parallel architects. However, It is sometimes difficult to use especially when the model becomes complicated or extremely small probabilities are used in the model. In this paper, we propose a reconfigurable Markov chain/queueing model simulation system. In this system, a user describes the target system in a dedicated description language called Taico. The description is automatically translated into the HDL description of Markov chain/queueing model simulator. Then, the simulator is implemented on FPGA devices of the reconfigurable system, and directly executed. From the evaluation with analysis of example parallel systems, it appears that the analysis speed of proposed system is hundreds times than that of common workstations.
key words: Markov chain, queueing model, reconfigurable machine

1 はじめに

離散時間マルコフチェーン (Descrete Time Markov Chain:DTMC) や待ち行列，ペトリネットは並列システムの動作解析や性能評価に広く用いられている．中でもDTMCと待ち行列は，数学モデルを専門としない並列システムの設計者にも，モデル化を含め習得が容易である．

しかしながら，実用的な規模の並列システムをモデル化した場合，DTMCは状態数の爆発的な増加により，待ち行列は多段化により，厳密な解析が困難となる．その場合，近似解法かシミュレーションによる解析が必要となるが，数学モデルを専門としない設計者にとっては，数学的知識を要する近似解法は適さず，シミュレーションによる解析が実用的な手段となる．しかし，複雑なシステムの信頼性解析などでは，発生確率の非常に小さな事象を扱うため，シミュレーションに多大な時間が必要となる．発生確率の小さな事象を高速にシミュレートするための方法は現在多く提案されているが [1][2]，いずれも数学的知識を必要とする．したがって，数学的な知識を必要とせず，解析対象のモデル化を行うだけで，高速にモデルのシミュレーションを実行できるツールが求められている．

一方，近年はField Programmable Gate Array (FPGA) やComplex Programmable Logic Device (CPLD) 等の，ユーザがプログラム可能なデバイスの発達により，解法アルゴリズムを問題に応じてハードウェア化し，プログラム可能なデバイス上で直接実行する可変構造マシン (Reconfigurable Machine)，あるいはCustom Computing Machineの研究が盛んになっている [3]．この方法は，専用マシンを開発する方法と比べ柔軟であり，解析対象に応じた構成をとることができることから，広い範囲の解析対象を扱うことができる．また処理速度の面においても，高速のワークステーション上でソフトウェアで実行した場合をはるかに上回ることができる [4]．

そこで本研究では，並列システムのDTMC／待ち行列モデルを，可変構造マシン上で高速にシミュレートするシステム，RSMS (Reconfigurable Stochastic Model Simulator) を提案し，設計と実装，および評価を行った．

RSMSのユーザは，次の手順で並列システムの解析を行う．

(1) 解析対象となるシステムを，DTMC または待ち行列ネットワークとしてモデル化する．

(2) 作成したモデルを，本システムの提供するモデル記述言語 Taico によって記述する．

(3) Taico によるモデル記述を，Taico-HDL トランスレータによってハードウェア記述言語 (Hardware Description Language: HDL) による記述に変換する．変換後得られる HDL 記述には，手順 (1) で作成したモデルをシミュレートする回路が記述されている．

(4) 得られた HDL 記述を論理／レイアウト合成し，FPGA を用いた可変構造マシン上にシミュレーション回路を実現する．

(5) 可変構造マシン上でシミュレーションを実行し，種々の計測データを収集する．

シミュレーション中のモデルの振舞いは，カウンタから構成された計測回路を通じて知ることができる．計測される項目は，DTMC に対しては，ユーザが指定した状態への到達回数と滞在時間，待ち行列ネットワークに対しては，各待ち行列の平均長やサービス完了数などである．

本システムのシミュレーション速度は，ワークステーション上でソフトウェアでシミュレーションする場合に比べて数十倍から数百倍高速であることから，ソフトウェアによる解析が時間的に困難な場合でも現実的な時間で解析を行うことが可能である．

以下，2章ではモデル記述言語 Taico について，3章ではシミュレーション回路のハードウェアアーキテクチャについて述べる．4章では実際の並列システムを解析対象として本システムの性能を評価する．

2 モデル記述言語 Taico

2.1 Taico による DTMC の記述

DTMC は，解析対象のとり得る状態と，状態間の遷移確率を図示する状態遷移図によって表すことができる．しかし並列システムでは複数の要素が相互作用しつつ動作するため，全体を一つの状態遷移図として表すよりも，個々の要素毎に状態遷移図を用意し，それらの集合としてシステムを表した方が便利である．このとき，全体の状態は個々の要素のとる状態の組で表されることになる．すなわち，並列動作する N 個の要素 $E_1, E_2, \cdots, E_N$ からなるシステムにおいて，要素 $E_i, 0 \leq i \leq N$ の状態を S_{E_i} で表すものとすると，システム全体の状態は N 個の組 $(S_{E_1}, S_{E_2}, \cdots, S_{E_N})$ で表される．Taico ではこの考

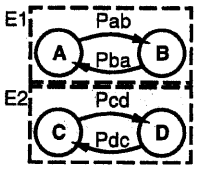

図 1: 2 つの要素からなる並列システムの状態遷移図

```
1   element E1 {
2      states A, B
3      A if(E2@D) : A
4        else     : B Pab
5                 | A 1-Pab
6                 ;
7      B if(E2@D) : B
8        else     : A Pba
9                 | B 1-Pba
10                ;
11  }
```

図 2: Taico による E1 の記述

えに基づき，並列システムの各要素について状態遷移を個別に記述する．なお，要素間の相互作用は，状態遷移に付随する条件として記述される．

例として，図 1 に示すような 2 つの要素 $E1, E2$ から構成される並列システムを考える．$E1$ は A, B の 2 状態をもち，A から B への遷移確率は Pab，B から A への遷移確率は Pba とする．また，$E2$ は C, D の 2 状態をもち，C から D への遷移確率は Pcd，D から C への遷移確率は Pdc とする．また，相互作用として，$E2$ が状態 D に存在する間は，$E1$ は状態遷移を行なうことができないものとする．

$E1$ の動作を Taico で記述したもを図 2 に示す．1 行目は $E1$ という要素名の宣言と，その状態遷移記述の開始を表す中括弧が書かれている．2 行目では要素のとる状態 A, B が宣言されている．3 ～ 6 行目は状態 A からの遷移記述であるが，3 行目において，$E2$ が状態 D にある場合 $E1$ は無条件に A に留まることが記述されている．また，そうでない場合は確率 Pab で状態 B に遷移する (4 ～ 6 行目)．7 ～ 10 行目は状態 B からの遷移記述である．また RSMS では，シミュレーション中に解析対象が特定の状態に (1) 何回到達したか，(2) 何ステップ滞在したかをカウンタによって計測することができる．計測の対象となる状態の指定は，(1) の項目については reach(状態を

```
1   source S {
2      Call_type   TypeA
3      Distr       M;
4      Param       0.1;
5   }
6   queue Q {
7      Call_source  S;
8      Max_length   Inf;
9   }
10  server SV {
11     Call_source  Q;
12     Serve_type   M;
13     Param        1.5;
14  }
```

図 3: Taico による待ち行列モデルの記述例

指定する式), (2) の項目についてはstay(状態を指定する式) で指定する. 上の例において $E1$ が何回 B に到達したかを計測したい場合, reach(E1@B) と記述する. この場合, $E2$ の状態に関わらず, $E1$ が状態 B に遷移するたびにカウンタが増される.

2.2 Taico による待ち行列の記述

並列システム等の複雑な解析対象は単一の待ち行列で表現されることはほとんどなく, 大抵は複数の待ち行列を相互に接続した待ち行列ネットワークとしてモデル化される. また, 呼についても幾つかの種類 (呼種) に分けてモデル化される場合が多い. 待ち行列ネットワークにおいては, あるサービス窓口 (サーバ) から退出した呼が別の待ち行列に新たな呼として到着したり, 他の呼と合流したりする. また, 呼の経路の分岐点おいては, 呼は確率的に経路を選択したり, 呼種に応じて経路が決定されたりする.

Taico では, 待ち行列ネットワークを記述するために, 呼の種類 (呼種), 呼源, 待ち行列, サーバ, 分岐点, 合流点, の各要素を個別に記述する. 図 3に, 呼源 S から待ち行列 Q に呼が到着し, サーバ SV でサービス受けた後退出するモデルの Taico による記述を示す. 1～5 行目は呼源の記述であり, 発生する呼の呼種は $TypeA$, 発生分布には母数 0.1 のポアソン分布が指定されている. 6～9 行目は待ち行列の記述であり, 呼源 Q から呼が到着し, 行列の長さは無制限であることが記述されている. 10～14 行目はサーバの記述であり, 待ち行列 Q にある呼に対し平均 1.5 単位時間の指数分布のサービスを提供することが記述されている.

RSMS において待ち行列ネットワークをシミュレートする場合, 呼の到着数, サービス完了数, サーバがサービスを行っている時間の平均, 待ち行列に存在する呼の平均数, 全シミュレーション時間が計測される. これらの計測はシミュレーション回路内に付加された加算器とカウンタによって実行される. これらの計測値によって, 解析対象の種々の性能指標値を計算することが可能である.

3 シミュレーション回路アーキテクチャ

3.1 可変構造マシン FLEMING

最終的なシミュレーション回路は, 可変構造マシン FLEMING (Flexible Logic EMulatIon eNGine)[5] 上に実現される. 図 4に FLEMING の構成を示す. FLEMING は, Xilinx 社の FPGA である LCA XC5215(15000 ゲート相当)[6] を 6 個, 制御用の 16bit マイクロプロセッサ, およびホストのワークステーションとの通信を行うインタフェースユニット (GPIB) から構成される. FLEMING は本来, 仮想ハードウェア WASMII[7][8] をエミュレーションする目的で開発されたシステムであるが, 本システムのようなマルコフチェーンシミュレータを実現するための, Reconfigurable Machine のテストベッドとしても利用することができる.

FLEMING では各 LCA に対し, コンフィグレーションデータ格納用のメモリ, および各種コントロール用の回路と補助目的のワークメモリが割り当てられており, これらをまとめて Reconfigurable Unit (RU) とよんでいる. ホストマシン上で生成されたコンフィグレーションデータは, 制御用プロセッサによりコンフィグレーションデータ用メモリに転送され, LCA 上でハードウェアが実現される. RU 間の接続は, Interconnection Unit (IU) を構成する EEPROM 型の配線用 FPGA の変更により, さまざまな形式をとることができる.

3.2 DTMC のシミュレーション回路アーキテクチャ

Taico による DTMC の記述では, 解析対象は並列動作する複数の要素に分割され, 個別に状態遷移が記述される. 記述されたモデルをシミュレートする回路は, 図 5に示すように, Taico による記述された個々の要素をステートマシンとして実現する. この図は N 個の要素からなる並列システムをシミュレートする場合の例であるが, 他の場合も要素数や乱数発生器の有無以外は同様のアーキテクチャとなる.

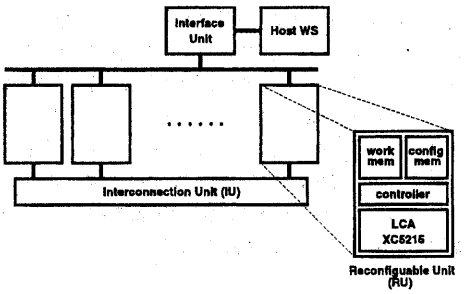

図 4: Reconfigurable Testbed FLEMING

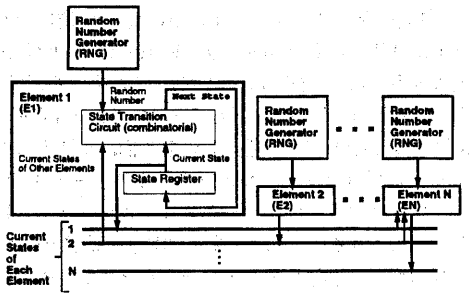

図 5: DTMC のシミュレーション回路アーキテクチャ

各要素は，現在の状態を記憶するレジスタと組合せ回路からなるステートマシンである．これらは，現在の自分の状態と，バスを介して受け取る他の要素の現在の状態，乱数発生器の生成する疑似乱数から次の状態を決定する．要素間の相互作用に応じたバスの接続や，確率的状態遷移を行う要素への乱数発生器の付加は，後述する Taico-HDL トランスレータによって行われる．

各ステートマシンはどれも共通のクロックで動作する．このクロックの 1 周期はシミュレートされる DTMC の 1 ステップに対応する．従って，この回路の最高動作周波数が，そのまま DTMC のシミュレーション速度 (steps/sec) となる．

3.3 待ち行列のシミュレーション回路アーキテクチャ

DTMC の場合と同様，待ち行列のシミュレーション回路は，Taico によって記述された呼源やサーバ等の要素を，それぞれの動作をシミュレートする回路に個別に置き換えたものとなる．回路中では，呼は単発のパルスとして表現される．例えば，呼源は指定された確率分布に従い，呼に対応するパルスを発生する回路によりシミュレートされる．DTMC の場合と同様，待ち行列のシミュレーション回路全体は共通のクロックで動作し，最高動作周波数が待ち行列のシミュレーション速度 (unit time/sec) となる．

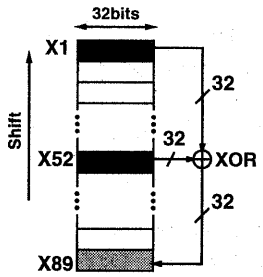

図 6: 疑似乱数発生器

3.4 Taico-HDL トランスレータ

Taico-HDL トランスレータは，Taico による解析対象の記述を，前節で述べたアーキテクチャを持つシミュレーション回路の HDL 記述に変換する．本システムでは HDL として Verilog-HDL を用いており，トランスレータ自身は C 言語で記述されている．

トランスレータは，Taico による各要素の記述を Velilog-HDL のモジュール記述として 1 対 1 に変換した後，トップモジュールの付加，および後述の計測ブロックや乱数発生器の結合を行う．

3.5 乱数発生器

シミュレーションにおける擬似乱数の役割は重要であり，さまざまな生成アルゴリズムが提案されている [9]．本システムでは，長周期の擬似乱数列をハードウェアによって高速に生成する必要があるため，M 系列による方法を採用している．用いた漸化式は

$$X_n = X_{n-89} \oplus X_{n-38} \quad (n \geq 90) \qquad (1)$$

であり，ビット長は 32 である．ここに $\oplus$ はビットごとの排他的論理和を表す．(1) 式により発生される疑似乱数列の周期は $2^{89} - 1$ である．図 6 に RSMS で用いる乱数発生器の構成を示す．この方法により，排他的論理和回路とシフトレジスタによる単純な構成の乱数発生器が実現できる．また，シフトレジスタの 1 動作クロック毎に乱数を生成できる．しかし一方で，32 ビット × 89 段のシフトレジスタが必要となり，回路規模は大きくなる．

4 評価

本章では，次の4つの並列システムを解析対象として本システムを適用した結果について述べ，本システムの性能を考察する．

(1) IEEE Futurebus のバスアービトレーションプロトコルの動作 (プロセッサ数 8)

(2) 4入力 MIN 結合型並列計算機におけるメモリアクセス動作

(3) 複数の共通バスで結合されたマルチプロセッサシステムのメモリ／バスアクセス動作

(4) 8入力 MIN 結合型並列計算機におけるメモリアクセス動作

モデル化に当たっては，(1),(2) を DTMC，(3),(4) を待ち行列ネットワークとしてモデル化した．

(1) はバス結合型マルチプロセッサにおけるアービトレーション動作であり，各プロセッサを並列動作要素として記述している．この場合各要素は6つの状態をもつ状態遷移図としてモデル化される．

(2) は4つのプロセッサが，代表的な多段接続網 (Multistage Interconnectin Network: MIN) であるオメガ網を通じ，4つのメモリモジュールに接続されたシステムである．オメガ網を構成するスイッチは2入力2出力のもので，直通接続と交換接続をとる．さらに，各スイッチはマルコフチェーンの各ステップにおいて一定の確率で故障するものとする．スイッチの個数は4である．モデル化にあたっては各プロセッサとスイッチを要素とみなし，各プロセッサは13状態，各スイッチは7状態の状態遷移図としてモデル化した．

(3) は2つのプロセッサのグループが，3本の共通バスを通じ，3個のバンクにインタリーブされた共通メモリにアクセスするモデルである．プロセッサはグループごとに異なる頻度でアクセスを行うものとする．モデル化にあたっては各プロセッサグループを呼源，各メモリバンクを待ち行列とサーバとしてモデル化している．

(4) は8つのプロセッサが，オメガ網を通じ8つのメモリモジュールに接続されたものである．オメガ網は (2) で述べたものと同じスイッチで構成されており，スイッチの個数は12である．モデル化にあたっては，各プロセッサを呼源，各スイッチを2つの待ち行列，各メモリモジュールを待ち行列とサーバとしている．

上記 (1),(2) については，単一の FPGA 内にシミュレーション回路が実現可能である．(3),(4) は複数の FPGA にまたがってシミュレーション回路が実現されるが，現時点では FLEMING 上で複数の FPGA にまたがってシミュレーション回路を実現する環境が整っていないため，(3),(4) についての評価は論理／レイアウト合成ツールの出力する遅延の計算値によって行った．

4.1 論理合成結果

上に述べた4つの並列システムの Taico による記述をトランスレータにより Verilog-HDL に変換し，論理／レイアウト合成した．なお，論理合成には MenterGraphics 社 Autologic，レイアウト合成には Xilinx 社 XACT を使用した．FLEMING で用いている XC5215-6 (15,000 ゲート相当 [6]) 上での最大動作周波数と基本セルである CLB(Configurable Logic Block) 数，換算ゲート数を表1に示す．FPGA を用いた場合，CLB の利用率によるゲート換算は正確なものではなく，目安にすぎない．なお，乱数発生器は回路が大規模であるため，状態遷移をシミュレートする回路とは別個の LCA チップ上に実現するものとして論理合成した．

表 1: 各モデルのシミュレーション速度およびハードウェアコスト

	最大動作周波数	基本セル使用数	換算ゲート数
Futurebus	9.1 MHz	345	10350
4入力 MIN	9.4 MHz	349	10470
8入力 MIN	11.6 MHz	837	25110
複数共通バス	15.3 MHz	302	9060
乱数発生器	3.2 MHz	243	7290

表1からわかるように，乱数発生器が全体の動作速度を低下させている．しかし，それでも1秒あたり300万ステップ以上のシミュレーションが可能である．これに対し，ワークステーション (Hewlett Packerd 715/64, PA-RISC 64MHz) において，同様のシミュレーションをソフトウェア (C 言語による) で行った結果を表2に示す．

この結果より，4入力 MIN 結合型並列計算機については 10^9 ステップのシミュレーションを行った場合，ワークステーションでは約9万秒 (約25時間) を要するのに対し，本システムでは動作クロック 3MHz の場合でも約330秒で済むことがわかる．これは論理／レイアウト合成に要する時間 (本例では約1時間) を差し引いても十分実用になる性能である．

表 2: ワークステーションのシミュレーション速度

	動作速度 (steps/s)
Futurebus	16,700
4 入力 MIN 結合網	11,000
8 入力 MIN 結合網	2,100
複数共通バス	5,900

4.2 FLEMING 上での実行結果

次に，上で論理／レイアウト合成したシミュレーション回路のうち，Futurebus と 4 入力 MIN 結合網について FLEMIMG 上で動作確認を行った．ただし，乱数発生器については回路規模が大きく，シミュレーション回路と同一の FPGA 内に実装することができなかったため，$32bit \times 1$ 段の簡易型 M 系列乱数発生器を HDL 記述により作成し，シミュレーション回路と同一の FPGA に内蔵させて動作確認を行なった．その結果，両者とも，8MHz のクロックでの動作を確認することができた．

4.3 乱数発生器の性能

乱数発生器では，32 ビット ×90 段のシフトレジスタがハードウェアの巨大化と速度低下の原因となっている．これは，本評価で用いた FPGA がメモリを内蔵していなかったため，シフトレジスタが効率良く実現できず，またその結果として動作速度が低下しているためである．しかし，この問題に対しては，メモリ内蔵型の FPGA (例えば Altera の FLEX10K シリーズ [13]) を利用することによりシフトレジスタを効率良く実現することが可能となり，性能改善が期待できる．

5 むすび

DTMC／待ち行列モデルを可変構造マシン上でシミュレートするためのシステムを提案し，実装と評価を行った．その結果，ワークステーションに比べ数十倍から数百倍の速度でシミュレーションが実行できることがわかった．

参考文献

[1] Perwez Shahabuddin: "RARE EVENT SIMULATION IN STOCHASTIC MODELS", Proc. of the 1995 Winter Simulation Conference, pp.178-185, 1995

[2] M., and J. Villen-Altamarino: "RESTART: a straightforward method for fast simulation of rare events", Proc. of the 1994 Winter Simulation Conference, pp.282-289, 1995

[3] 沼昌宏: "FPGA を利用したアーキテクチャとシステム設計", 情報処理, Vol.35 No.6, pp.511-518, Jun. 1994.

[4] M.Gokhale, et al.: "Building and Using a Highly Parallel Programmable Logic Array", Computer, Vo. 24, No.1, pp.81-89, 1991.

[5] Y.Shibata, X-P.Ling, H.Amano: "An Emulation System of the WASMII: Data Driven Computer on a Virtual Hardware," Proc. of FPL'96, (LNCS 1142), pp.55-64, 1996.

[6] XILINX Corp.: Programmable Gate Array's Data Book July 1996

[7] 凌 暁萍、天野 英晴、"WASMII: データ駆動型制御機構を持つ MPLD," 電子情報通信学会論文誌 Vol.J77-D-1, No.4 1994 年 4 月

[8] X-P.Ling, H.Amano : "WASMII: AN MPLD with Data-Driven Control on a Virtual Hardware," The Journal of Supercomputing, 9, 253-276, 1995.

[9] Pierre L'Ecuyer: "RECENT ADVANCES IN UNIFORM RANDOM NUMBER GENERATION", Proc. of the 1994 Winter Simulation Conference, pp. 176-183, 1994

[10] "IEEE Standard Backplane Bus Specification for Multiprocessor Architectures: Futurebus," IEEE Jun. 1988.

[11] 鳥居 淳, 天野英晴: "並列計算機テストベッド ATTEMPT の交信機構の評価", 並列処理シンポジウム JSPP '91 論文集, pp.205-212, May. 1991.

[12] Takuya Terasawa, O Yamamoto, Tomohiro Kudoh, Hideharu Amano: "A performance evaluation of the multiprocessor testbed ATTEMPT-0", Parallel Computing 21, pp.701-730, 1995

[13] ALTERA: Data Book Jun. 1996.