オペレーティング・システム 50−11
（1991. 3. 16）

マルチプロセッサシステムのための性能評価ツール MUSES

鴨川 郷　秦泉寺 浩史　山本 幹　岡田 博美　中西 暉　手塚 慶一

大阪大学工学部

並列処理システムの代表的な一形態であるマルチプロセッサシステムの性能は，アーキテクチャ等のハードウェア構成及びその上で実行される並列アルゴリズム等のソフトウェアの構造の双方に大きく依存する．本稿では，ソフトウェア記述により構築した仮想マシン上のインストラクションレベルでのシミュレーション，及び確率モデルを用いたモンテカルロシミュレーションを融合することにより，大規模マルチプロセッサシステムに対しハードウェアとソフトウェアの両者の影響を総合的に評価できる性能評価ツール，ＭＵＳＥＳ(MUltiprocessor System Evaluator by Simulation)を提案し，その有効性を検証する．

Performance Evaluation Tool for Multiprocessor System −MUSES−

Akira Kamogawa　Hiroshi Jinzenji　Miki Yamamoto　Hiromi Okada　Hikaru Nakanishi　Yoshikazu Tezuka

Faculty of Engineering, Osaka University

2-1 Yamadaoka Suita-shi Osaka 565 Japan

Performance of multiprocessor systems depends on both its hardware architecture and software executed on it. In this paper, we propose a performance evaluation tool for multiprocessor systems, MUSES(MUltiprocessor System Evaluator by Simulation). MUSES integrates simulation at instruction level and simulation which uses stochastic models of instruction generation and accesses to the interconnection network. MUSES can evaluate large scale multiprocessor systems precisely, taking into account of the interaction of software and hardware.

1．まえがき

現在，ハードウェア装置の高性能化，多機能化と，多様な要求に対処できる膨大なソフトウェアの蓄積によって，データベース処理，文章処理，図形・画像処理，音声処理などの様々なデータ処理分野にコンピュータが利用されている．このような応用範囲の拡大にともない，より高性能なコンピュータシステムの開発が切望されている．

コンピュータシステムの性能，特に処理速度の向上は，これまで，システムの各素子自体の処理速度を向上させる素子技術の発展によって支えられてきた．しかし，素子技術の発展による高速化には物理的に限界があり，素子技術の発展だけでは今後の飛躍的な性能向上は望めない段階に来ている．コンピュータシステムの今後のさらなる性能向上に対しては，現在までのシステム構成とは全く異なる新しいシステムアーキテクチャが必要となる．このような状況において，複数の処理装置で並列に処理を実行し，処理の高速化を図る，並列処理システムに対する研究が注目されている．本来，並列処理は問題に含まれる同時実行可能な部分を抽出し，それを小問題に分割し，複数の処理装置を用いてそれらを並列処理することで処理の高速化を図る．この並列処理を行う目的で開発されたシステムの1つに，問題の中から同時に実行できる処理を様々な段階から見つけて，それらを並列処理するマルチプロセッサシステムがある[1]．このマルチプロセッサシステムは，高い信頼性，拡張性，適用範囲の広さという特徴を持ち，高い能力を持つコンピュータシステムを実現する上で今後有望なシステム構成であると考えられている．

このマルチプロセッサシステムの研究課題は，アーキテクチャ等のハードウェアに関する問題，並列アルゴリズム等のソフトウェアに関する問題の2つの分野に分類できる．これまで，マルチプロセッサシステムの性能向上のために，ハードウェアとソフトウェアのそれぞれの研究課題に対して，様々な研究がなされ，各研究課題に対する最適な方式が模索されている．しかしながら，これらの研究においてはハードウェアに関する研究課題とソフトウェアに関する研究課題を独立に取り扱っており，ハードウェアとソフトウェアの相互影響が観測されるマルチプロセッサシステム全体に対する性能評価は行っておらず，最終的な解決策を得る段階には到っていない．つまり，高性能で実用的なマルチプロセッサシステムの開発のためには，ハードウェアに関する問題，ソフトウェアに関する問題の総合的な評価が必要となり，この評価を可能とする性能評価ツールが不可欠となる．

従来，マルチプロセッサシステムの性能評価には，待ち行列網を用いた近似解析，またはモンテカルロシミュレーションがよく用いられてきた[2][3][4]．この方法では，実システムから評価の対象となる要素または機能のみを抽出し，その変化規律を確率分布等でモデル化し，システムの動作を模倣することで性能評価を行う．しかし，この手法では，ソフトウェア実行時における通信路およびメモリでのアクセス競合等の，その状態変化が過去の履歴，及び現在の状態に大きく依存するために，簡単な確率モデルで表現できない事象に対する評価は不可能である．

また，従来のもう1つの性能評価方法に，目的のシステムを計算機上に仮想マシンとして構築し，インストラクションレベルでシミュレートする方法[5][6]がある．近似解析並びにモンテカルロシミュレーションがシステムを抽象的にモデル化しているのに対し，この方法では，目的のシステムと同様の機能を有する仮想マシンを構築し，抽象レベルの非常に低いところでマシンの動作を模倣し，統計量等所望のデータを抽出するものである．しかし，この方法では，多数のＰＥが存在し，その上で複数のジョブが実行されている場合には仮想マシン上で実行させる並列プログラムの記述に多大な労力を必要とする．このため，大規模システムの性能評価は困難となる．

以上の検討から，ハードウェアだけでなくソフトウェアの影響をも考慮したマルチプロセッサシステムの性能評価手法として，モンテカルロシミュレーションと，仮想マシン上でのインストラクションレベルのシミュレーションとを，融合する方法が考えられる．これは，インストラクションレベルのシミュレーションによりハードウェアとソフトウェアの相互影響を加味しつつ，大規模システムへの対応をモンテカルロシミュレーションによりはかるというアプローチである．

本稿では，仮想マシン上のインストラクションレベルでのシミュレーション，及び確率モデルを用いたシミュレーションを融合することにより，大規模システムに対しハードウェアとソフトウェアの両者の影響を総合的に評価できる，ＭＵＳＥＳ(MUlti-processor System Evaluator by Simulation)について述べる．

２．ＭＵＳＥＳの概要

ＭＵＳＥＳは，複数のプロセッサエレメント（ＰＥ）と共有メモリが相互結合網で結合されており，この複数のＰＥが相互結合網を介して共有メモリにアクセスしながら，協調して仕事を行う共有メモリ型マルチプロセッサシステム形態を評価対象とする．

さらに，ＭＵＳＥＳが評価対象とするマルチプロセッサシステムは以下の特徴を有するものと仮定する．

・各プロセッサエレメントはローカルメモリを有している．
・各プロセッサエレメントに与えられるユーザプログラムは，ローカルメモリに格納されている．
・各プロセッサエレメントはキャッシュを有さない．
・プロセッサ間通信は共有メモリを介して行われる．

本研究で開発するＭＵＳＥＳにおいて，実行されるシミュレーションを分類すると，以下のようになる．

＜インストラクションシミュレーション＞

このシミュレーションはインストラクションを仮想マシン上で実行することによるシミュレーションである。このシミュレーションはその性質によって２つに分類できる。

・並列アルゴリズムに基づくインストラクションシミュレーション

このシミュレーションにおいてはユーザの与えたプログラム（並列アルゴリズム）により決定される順序でインストラクションを実行する．これにより，並列アルゴリズム実行時の詳細はシミュレーションが可能となる．

・確率分布に基づくインストラクションシミュレーション

このシミュレーションはインストラクションをある確率分布に基づいて発生させ、そのインストラクションをユーザプログラムに基づくインストラクションと同様の方法で実行する。このシミュレーションは多数のＰＥを有する大規模マルチプロセッサシステムの性能評価をするために用意されている。

＜イベントシミュレーション＞

このシミュレーションは２つに分類できる。

・ＰＥイベントシミュレーション

これは指数分布等のある確率分布に基づいてＰＥから発せられるメモリアクセスをシミュレーションする。これも多数のＰＥをシミュレーションするものである。

・外部要因イベントシミュレーション

これはＩ／Ｏ等のＰＥブロック、ＣＯＭブロック及びＭＥＭブロックに対する影響をシミュレートするために用意されている。

性能評価ツールに一般に要求される条件に，①評価結果の精度，②幅広い評価対象を網羅する柔軟性，が挙げられる．ＭＵＳＥＳでは，①に対しては並列アルゴリズムに基づくインストラクションシミュレーションによって，ソフトウェアの影響も加味した精度の高い性能評価が行えるよう対応している．ＭＵＳＥＳにおいては，ＰＥでのインストラクションの実行，ＰＥ及び共有メモリから相互結合網へのアクセス，及び共有メモリに対する読み出し，書き込み動作等のマルチプロセッサシステムの動作をソフトウェアで記述し実行させることにより，ソフトウェア記述による仮想マシンを構築している．ユーザが与える並列アルゴリズムを仮想マシンであるマルチプロセッサシステム上で実行させた場合の順序にしたがい，これらの動作を実現することにより，目的のシステム上で並列アルゴリズムが実行されているのと同一の統計データが抽出できる．これによって，対象となるマルチプロセッサシステム上で実行させる並列アルゴリズムの影響を考慮した精度の高い性能評価が可能となる．②への対応の前半部に対しては，その他の３種類のシミュレーションを併用することによって，大規模システムへの適用性を改善している．②への対応の後半部に対しては，仮想マシンであるマルチプロセッサシステムの基本構成要素であるプロセッサエレメント（ＰＥ），相互結合網（ＣＯＭ），共有メモリ（ＭＥＭ）のそれぞれに対し，その機能を実現するモジュールを構築し，評価

対象のマルチプロセッサシステムを変更した場合，機能モジュールの変更で対応できるようにすることで解決をはかっている．また，確率モデルを用いたシミュレーションに対しては，別に，モジュール（ＳＩＭ部）を構築し，実行する．

以上，ＭＵＳＥＳはソフトウェアとハードウェアの双方の影響を加味しつつ，マルチプロセッサシステムに対する精度の高い性能評価を柔軟に行える性能評価ツールである．

３．ＭＵＳＥＳの構成

３．１．システム構成

ＭＵＳＥＳは図１に示すように，プロセッサエレメント（ＰＥ）部，コミュニケーション（ＣＯＭ）部，メモリ（ＭＥＭ）部，シミュレーション（ＳＩＭ）部，システムモニタ部からなる．ＰＥ部には仮想マシン上に存在するＰＥの数に等しいだけのブロックが用意され，各ブロックが仮想マシン上の各ＰＥに対応する．共有メモリ型マルチプロセッサシステムにおいて，各ＰＥが他のＰＥに与える影響は相互結合網を通してのみ伝わるという特徴を有する．従って，ＭＵＳＥＳにおいては，相互結合網を通しての通信機能を実現するＣＯＭ部を介して，各ＰＥに対応するＰＥ部のブロックが影響を及ぼし合うような構造をとっている．更に，大規模システムでの多数のＰＥからの通信路アクセスをモデル化するＳＩＭ部がＣＯＭ部を介して各ＰＥに影響を及ぼすという形をとる．また，ＭＵＳＥＳは評価対象である仮想マシンのクロックを基本クロックとして時刻駆動される．すなわちＭＵＳＥＳの各部はシステムモニタ部の発する基本ｃｌｋによってｃｌｋ同期されている．

ＭＵＳＥＳの制御ルーチンは図２のようになる．図２に示すように，ＭＵＳＥＳが起動されると各フラグ，及びローカルメモリが初期化され，さらにユーザの与えたファイルに基づき共有メモリが初期化される．その後，ユーザプログラムのファイルに基づき，そのインストラクションを実行する．インストラクションの実行中は，ｃｌｋ毎に制御ルーチンのループ部分を実行する．アクセス処理の関係上，ＰＥ部，ＣＯＭ部，ＭＥＭ部の各部の処理は最初にＣＯＭ部，次にＭＥＭ部，最後にＰＥ部の順序で行われる．ユーザプログラムの終了等の各ブロックの終了条件を全てのブロックが満たすと，制御ルーチンのループ部分を出て，統計データを出力し，終了する．

以下の節において，ＭＵＳＥＳを構成する各部についてさらに詳しく述べる．

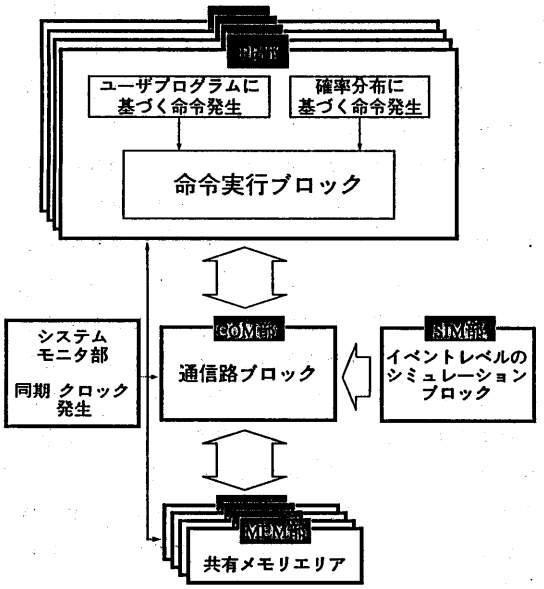


図１．システム構成

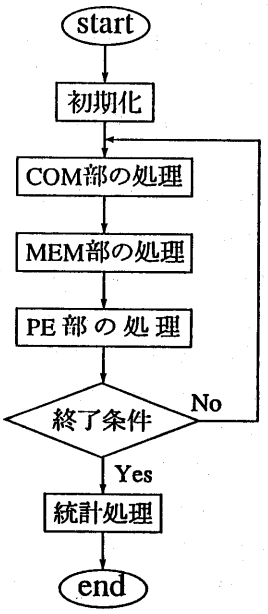


図２．ＭＵＳＥＳの制御ルーチン

3．2．プロセッサエレメント部

本節においては，ＭＵＳＥＳにおけるプロセッサエレメント部（以下ＰＥ部）の動作について詳しく説明する．

マルチプロセッサシステムにおける各ＰＥの命令実行形態を図３に示す．各ＰＥは，本来，図３に示す動作を連続的に実行するが，ＭＵＳＥＳにおいては，これらの状態を離散的にとらえ，状態変化は状態遷移時点に集約させて，この時点において，その状態内での動作をすべて実現するという形態をとる（例えば，命令実行状態に遷移する際には，この状態に遷移する時点において命令実行動作をすべて実現する）．ところが，ＭＵＳＥＳは評価対象マルチプロセッサシステムの１クロックごとに処理を実行するので，ＰＥが何も処理しない状態も存在することになる．例えば，対象マルチプロセッサシステムのＰＥにおいて命令実行に５ｃｌｋ要する場合には，ＭＵＳＥＳのＰＥでは最初の１ｃｌｋで全処理を終了してしまうので，残りの４ｃｌｋは何も処理を行わない状態となる（ｃｌｋ調整）．このことも踏まえて，ＭＵＳＥＳにおいては，インストラクションレベルのシミュレーションを実行するＰＥの状態を図３内の命令実行形態に基づき，以下の８つに分類する．

① 実行前処理
② 実行前処理のｃｌｋ調整中
③ ｒｅａｄアクセス要求発生
④ ｒｅａｄアクセス待ち
⑤ ＰＥ内部処理
⑥ ＰＥ内部処理のｃｌｋ調整中
⑦ ｗｒｉｔｅアクセス要求発生
⑧ ｗｒｉｔｅアクセス待ち

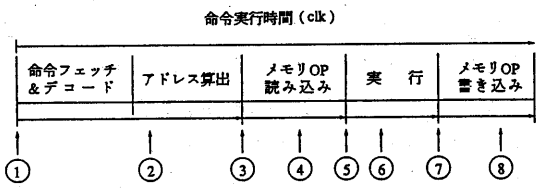
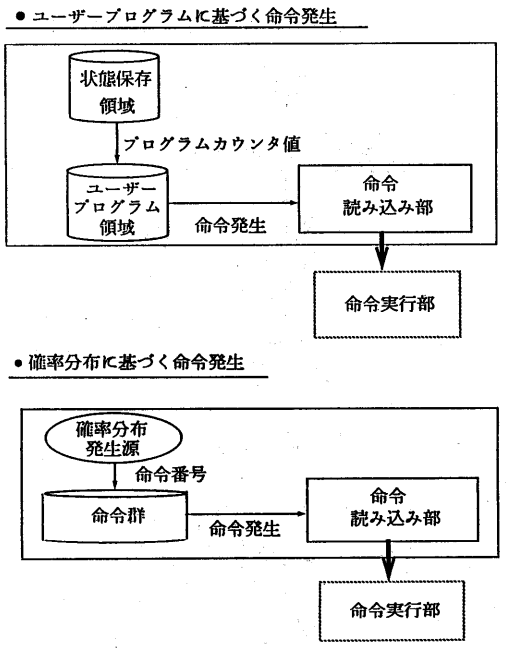


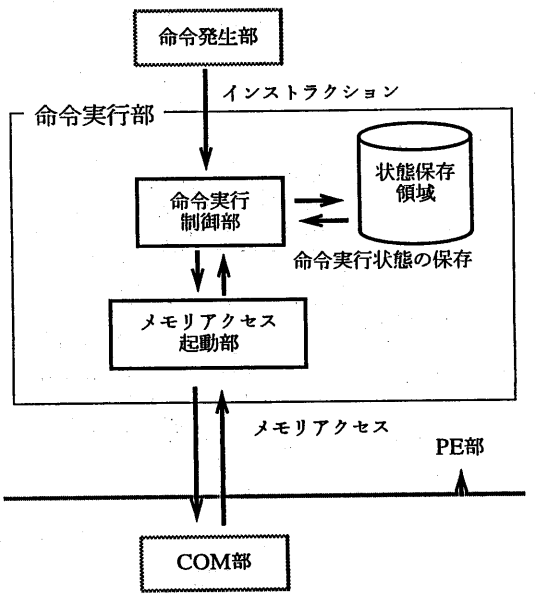
図３．命令の実行形態

プロセッサエレメント部の制御法

プロセッサエレメント部の構成は図４（ａ）（ｂ）の通りである．図３の命令の実行形態に基づき，①の状



（ａ）命令発生部



（ｂ）命令実行部

図４．プロセッサエレメント部

態にあるＰＥにおいて命令発生部が起動される．ユーザープログラムに基づく命令発生では，状態保存領域からそのときのプログラムカウンタの値がユーザープログラム領域に引き渡され，そのプログラムカウンタの値にしたがった新しい命令が命令実行部に渡される．確率分布に基づく命令発生では，ある確率分布にしたがった命令の番号が命令郡に引き渡され，その番号にしたがった新しい命令が命令実行部に渡される．命令実行部では，各クロックでのＰＥの状態を状態保存領域に保存しながら命令を実行し，メモリアクセスの際には，ＣＯＭ部を通じてメモリにアクセスする．

３．３．コミュニケーション部

コミュニケーション部の制御法

図５にコミュニケーション部（ＣＯＭ部）の構造を示す．コミュニケーション部はＰＥ部とＳＩＭ部からのアクセス要求を受け取るインターフェース，その時のＣＯＭ部の状態を保持している状態保存領域，競合解決法を提示する競合解決部，これらの各部を制御する制御部から構成されている．各クロックにおいて，以下の順序でＰＥ部，ＳＩＭ部，ＭＥＭ部からの通信要求を処理する．

Ⅰ．新しくアクセス要求を発したＰＥ部を調べる．

Ⅱ．現在進行中のアクセスの内，終了したアクセスを調べる．

Ⅲ．現在のアクセスの状況を考慮し，相互結合網の競合解決の方法にしたがって，相互結合網及びメモリの競合を解決する．

Ⅳ．通信路を選択する．

Ⅴ．そのクロックでの相互結合網の状態を決定する．

Ⅵ．選択された経路をＰＥ部及びＭＥＭ部に知らせる．

３．４．メモリ部

メモリ部の制御法を以下に示す．

コミュニケーション部において経路が選択され，決定すると，メモリ部は図６のルーチンを実行する．各メモリ部はまず，コミュニケーション部から，自分と接続されているＰＥに関する情報を得る．つぎに，自分と接続されているＰＥのアクセス要求を確認し，そのアクセス要求の種類（READ or WRITE），アクセスすべきアドレス，及びアクセス要求が WRITE のときには書き込みデータのそれぞれを決定する．その後，その要求に従った処理を行い，その処理結果をコミュニケーション部に知らせる．この後，アクセスが終了したときにコミュニケーション部にアクセスの終了を知らせる．

なお，ＭＵＳＥＳにおいては，共有メモリはメモリインターリーブされているものと仮定している．

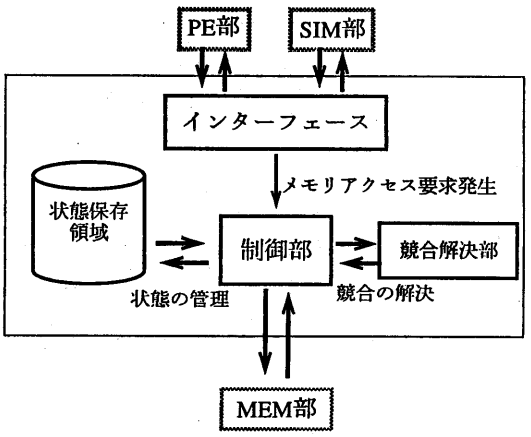


図５．コミュニケーション部

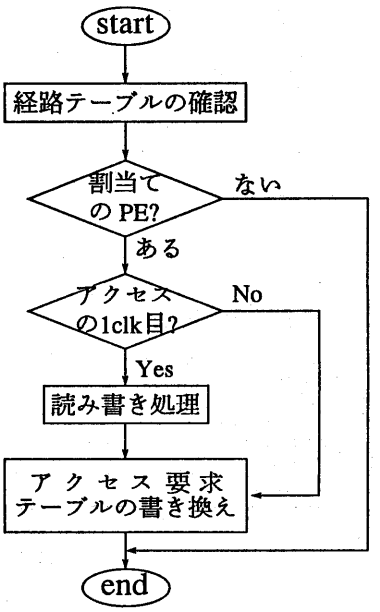


図６．ＭＥＭ部の制御ルーチン

3．5．シミュレーション部

シミュレーション部は、前章で示したシミュレーションの内、イベントシミュレーションを行うブロックである。シミュレーション部の構造は図7の通りである．2種類のイベントとも幾何分布等の確率分布に従って，それぞれのイベントを発生し，その発生したイベントに基づき，ＣＯＭ部にアクセス要求を発する．

4．評価例

ＭＵＳＥＳから得られる統計データとしては，次のようなものが挙げられる．

ＰＥ部で収集される統計データは以下に示すものである．

ＰＥ部の統計データの収集は1つのインストラクション終了ごとに行われる．

・各ＰＥで実行されたプログラムの文字列
・各ＰＥで実行されたインストラクションの種類
・各ＰＥで使用されたオペランドの種類
・インストラクションを実行するのに要したｃｌｋ数
・アクセスに要したｃｌｋ数
・アクセスの有無
・オペランドのアドレス算出時間
・現インストラクション終了までの所要時間

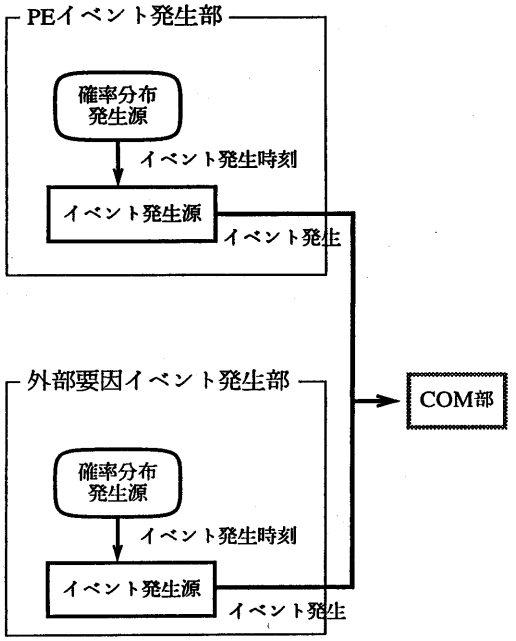


図7．シミュレーション部

ＣＯＭ部で収集される統計データは，以下に示すものである．

・各ｃｌｋにおける相互結合網を使用しているＰＥの番号

ＭＥＭ部で収集される統計データは，以下に示すものである．

・ＭＵＳＥＳが終了したときの，共有メモリの内容
・各ｃｌｋにおける各メモリモジュールとアクセスしているＰＥの番号

また，アルゴリズムに基づくシミュレーションが終了すると，ユーザプログラムの全実行時間を算出する．

ＭＵＳＥＳをＳＵＮ４ワークステーション上にＣ言語で構築し，マルチプロセッサシステムの評価例として，32個のデータに対する並列クイックソーティングを行った．その実行結果を表1に示す．ただし，シミュレーションとして，Ａ．４台のＰＥによる並列プログラムに基づくインストラクションシミュレーション，Ｂ．平均10clkの幾何分布に基づくアクセス要求を発生するイベントレベルシミュレーション，Ｃ．予め用意しておいた基本的なインストラクションから確率的にインストラクションを発生するインストラクションレベルシミュレーションを行った（Ａ－Ｃの記号は表の記号と対応する）．Ａのみの評価から並列アルゴリズムの実行時のシステム性能が得られ，Ｂ，Ｃとの組み合わせにより対象システムの拡張が容易に行われており，ＭＵＳＥＳがハードウェア並びにソフトウェア双方に関連した問題に対する総合的な評価を可能とし，大規模システムへの適応も可能としていることを確認した．

表1．クイックソーティングによる評価例

相互結合網	シミュレーションの種類	総アクセス時間	平均アクセス時間(clk)	プログラム実行時間(clk)
Ｂｕｓ型	Ａのみ	4680	4.05898	58090
Ｂｕｓ型	ＡとＢ	7968	6.91067	60110
Ｂｕｓ型	ＡとＣ	5386	4.67129	58555
Crossbar型	Ａのみ	4657	4.03903	58079
Crossbar型	ＡとＢ	5715	4.95663	58761
Crossbar型	ＡとＣ	5077	4.4033	58347

5．むすび

並列プログラムを用いたソフトウェアシミュレーションとモンテカルロシミュレーションを効果的に統合することにより，性能評価精度の高さと評価対象に対する柔軟性の双方をかね合わせたマルチプロセッサシステム性能評価ツールＭＵＳＥＳの開発を行った．また，各種の相互結合網によるマルチプロセッサシステム上でソーティング問題を実行し評価を行い，その有効性を検証した．現在，マルチプロセッサ型ＤＳＰシステムの評価のために，本評価ツールの改良を行っている．また，ソフトウェアシミュレーションで得られた評価結果を用いてモンテカルロシミュレーションのパラメータの調整を行うトレースドリブン機能の付加などが今後の課題である．

[参考文献]

[1] K.Hwang, F.A.Briggs : Computer Architecture and Parallel Processing, Mcgrow-Hill (1984)

[2] P.Heidelberger : "Computer Performance Evaluation Methodology", IEEE Trans. on Comput., vol.C-33, pp.1195-1220 (1984)

[3] S.S.Lavenberg : Computer performance Modeling Handbook, Academic Press (1983)

[4] D.Towsley : "Approximate Models of Multiple Bus MItiprocessor Systems", IEEE Trans. on Comput., vol.C-35, No.3 (1988)

[5] J.M.Butler, A.Y.Oruc : "EUCLID : AN ARCHITECTURAL MULTIPROCESSOR SIMULATOR", Proc.6th Int'conf. on Distributed Computer System, Boston , pp.280-287 (1986)

[6] J.M.Butler, A.Y.Oruc : "A Facility for Simulating Multiprocessors", IEEE MICRO, pp.32-44 (1986)