密結合マルチプロセッサシステムの性能評価に好適なシミュレーションモデルの提案†

藤井哲彦†† 根岸和義†† 米田茂††

密結合マルチプロセッサ (TCMP) システムの性能評価に関しては，各種の性能評価ツール，性能評価手法が提案されている．本論文では，トランザクションの応答時間の増加と1トランザクションの処理に要するCPUオーバヘッドの増加という2つの性能要素に着目し，TCMPシステムの性能を定量評価可能なシミュレーションモデルを提案する．提案するシミュレーションモデルは，ディスパッチャのモデル化とロック競合のモデル化を特に詳細に行うことにより，トランザクションの応答時間増加と1トランザクションの処理に要するCPUオーバヘッド増加の主たる性能要因をモデル化している．割込みや，タスクのディスパッチ遅れによるロック保有時間の増加をシミュレート可能であり，さらに空間およびタスクのキューサーチのシミュレートからキューサーチ時間を求めることにより，ディスパッチオーバヘッドとキューを排他制御するロックの競合オーバヘッドを正確に評価可能であるという特長を持っている．この特長によりトランザクションの応答時間の増加と1トランザクションの処理に要するCPUオーバヘッドの増加を定量的に評価することができる．ロックの競合度についての実測値との比較では，シミュレーション値と実測値が良く一致した．提案したシミュレーションモデルによりTCMPシステムの定量的な性能評価が可能となった．

1. はじめに

近年，大型汎用計算機は密結合マルチプロセッサ(TCMP)により，CPUパワーの増強をはかる方法が主流になっている．メインフレームメーカ各社から4～8台の構成が可能な機種が発表されている[1]．TCMPはプロセッサ台数に比例した性能を達成することが理想であり，われわれも高性能なTCMP制御方式の研究開発に取り組んでいる[10)~12)]．TCMPシステムでは現実にはリソースの競合などさまざまな要因により，プロセッサの多重度が大きくなるほどプロセッサ台数に比例した性能を達成することが難しくなる．高性能なTCMPシステムを実現するために，TCMPシステム内部の動的な挙動を定量的に把握し，的確な処理方式の開発に資する性能評価ツールが不可欠である．TCMPシステムの性能評価手法として，実時間のハードウェアトレースデータを用いてアーキテクチャの評価を行うシミュレータ[2]や，命令トレースデータを用いて階層記憶の評価を行うシミュレーション手法[3]，解析的な評価手法[4)~6)]などが知られている．実際にソフトウェアを設計する際には，現実のシステムを想定して，現状のソフトウェアでどの程度の性能が達成できるのかを予測し，性能ネックを特定し，そして処理方式を改良すればどのくらい性能が良くなるのかを評価することが必要になる．待ち行列を用いた手法や解析的な手法は，短時間に粗い精度での評価が可能であるが，上述したような，実システムを想定した正確な性能評価を行う際には，詳細なモデル化が難しく，あまり適さない．この場合，シミュレーションによる評価が適切であり[9]，コンピュータシステム全体を評価対象とするシミュレータとして，PACSS[7]やPRIME 3[8]が知られている．大型汎用計算機は，TSSサービスやバッチ処理を行う大学の計算センタや企業の開発システム，銀行システムのオンラインシステムなど，多様な形態がある．ここでは，より高い処理性能が求められるオンラインシステムに限定してTCMPシステムを考えていくことにする．TCMPシステムは一般にプロセッサの多重度が高くなるにつれて，プロセッサ1台あたりの処理性能が低下する．それはプロセッサ1台あたりの実効速度の低下，トランザクションの応答時間の増加，1トランザクションの処理に要するCPUオーバヘッドの増加という現象としてとらえることができる．

(1) プロセッサ1台あたりの実効速度の低下

プロセッサの多重度が上がると，メモリアクセスの競合，キャッシュヒット率の低下，パイプライン処理の乱れなどが原因となって，プロセッサ1台あたりの実効速度が低下してくる．

† Proposal of a Simulation Model for Evaluating Performance of Tightly-Coupled Multi Processor System by TETSUHIKO FUJII, KAZUYOSHI NEGISHI and SHIGERU YONEDA (6th Research Department, Systems Development Laboratory, Hitachi, Ltd.).

†† (株)日立製作所システム開発研究所第6部

（2） トランザクションの応答時間の増加

共通リソースの待ち時間の増加や，共通テーブルのアクセスを逐次化するために用いられるロックの競合による待ち時間の増加などによりトランザクションの応答時間が増加する．オンラインシステムでは，応答時間に許容限界値があり，許容値を超える場合には，負荷を制限しなければならない．したがって，トランザクションの応答時間の増加は，システムの処理能力の低下につながる．

（3） 1トランザクションの処理に要するCPUオーバヘッドの増加

共通リソースアクセスの競合度が増すことによる排他制御処理ステップの増加や，各種キューの構成要素数が増加することによるキュー処理ステップの増加などにより，1トランザクションの処理に要するCPUオーバヘッドが増加する．

以上にあげた原因により，プロセッサの多重度が高くなると，プロセッサ1台あたりの処理性能が低下する．TCMPのシステム性能を評価するには上にあげた3つの性能要素を総合して評価する必要がある．しかし，OS, DB/DCの排他制御や，プロセス管理等のソフトウェア上の性能要素を細かくモデル化し，ソフトウェアを性能評価の主眼としたTCMP性能評価シミュレータはこれまでに見当らない．本論文では上記(2),(3)に焦点を合わせ，応答時間増加とCPUオーバヘッド増加のメカニズムを詳細にモデル化し，TCMPシステム全体の性能評価を可能とするシミュレーションモデルを提案する．

2. シミュレーションモデル

本章で，評価対象システムの前提条件，シミュレーションの方針，シミュレーションモデルの構成，シミュレーションモデルの特長と効果について述べる．

2.1 評価対象システムの前提条件

評価対象システムの前提条件を以下に列挙する．

（1） 1種類のトランザクションだけがホストシステムに入ってくる1トランザクションモデルを仮定する．

（2） I/O時間はI/Oの種別ごとに常に一定とする．

（3） 多重空間，多重タスクシステムを仮定する．ディスパッチングキューは空間キューと空間ごとのタスクキューの2段階あるものを仮定する．ディスパッチ単位はタスクとアクションの2種類あり，アクションはノンプリエンプティブモード，タスクはプリエンプティブモードで実行される．アクションはタスクより高いプライオリティで実行され，IO処理の一部や，空間間の連絡のためのPOST処理等に使用される．タスクディスパッチ時には，プロセッサはまず空間キューをサーチして，レディタスクのある空間を見つけ，次に，その空間のタスクキューをサーチして，レディタスクを見つけディスパッチする．全空間中でレディタスクが見つからない場合にはプロセッサはアイドルウェイトする．ディスパッチングキューの構成とディスパッチャの動作を図1に示す．

（4） オンラインシステムは，システムで1つのDC (Data Communication) 空間と，メッセージを処理するための，システムに複数あるユーザ空間から構成される．DC空間には，トランザクションの振り分けを行うシステムタスクと，トランザクション処理を行う複数個の処理タスクがある．ユーザ空間には，空間ごとに1つずつユーザタスクがある．端末から入ってきたトランザクションはDC空間のシステムタスク，処理タスク，ユーザ空間のユーザタスクが連携して処理を行う．オンラインシステムの空間・タスク構成とトランザクション処理の流れを図2に示す．

（5） ロックはOSのスピンタイプロックとOSのサスペンドタイプロック，DB/DCのロックの3種類あるものとする．スピンタイプロックはロック確保失敗時にプロセッサがロック確保の試行を繰り返しながら（これをスピンすると呼ぶ）ロックが空くまで待ち

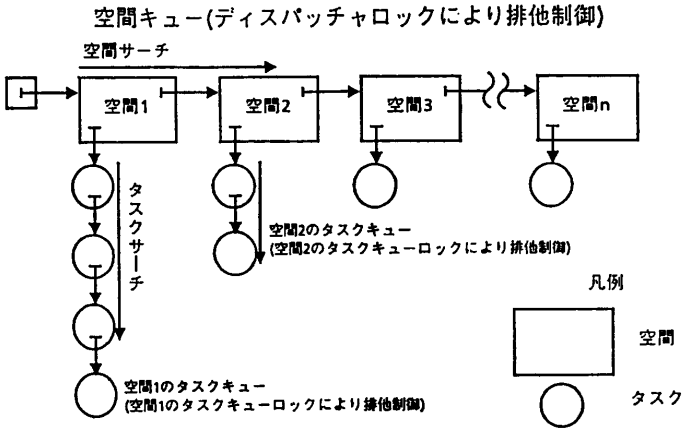

図1 前提とするディスパッチャ
Fig. 1 Premise of dispatcher.

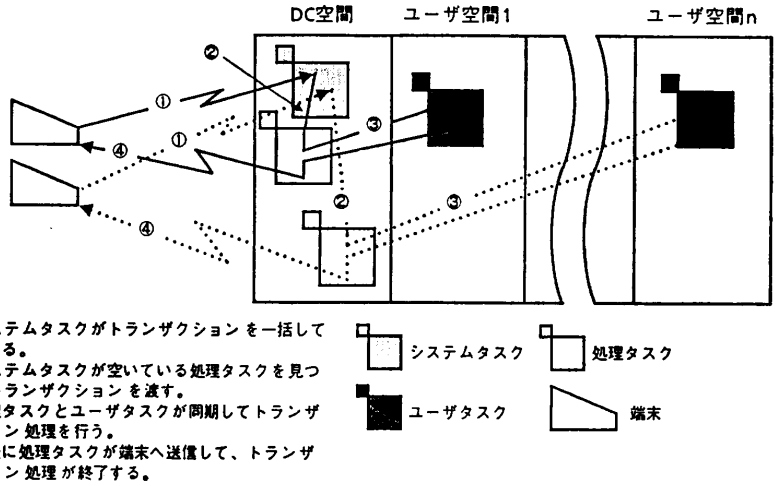

図 2 前提とするオンラインシステム
Fig. 2 Premise of online system.

続ける方法である．サスペンドタイプロックの制御は次のように行われる．タスクがロックの確保に失敗すると，そのタスクをサスペンド状態にさせ，他のタスクに制御を移す．ロックマネージャはFIFOのロック待ちタスクのキューを持っており，ロックが空き状態になった時，ロック待ちタスクがあれば，先頭のタスクのサスペンド状態を解き，ロックを確保させる．DB/DC のロックは次のように制御される．タスクがDB/DC のロック確保に失敗すると，制御はタスクからディスパッチャに移される．このときタスクはランニング状態からレディ状態となり，次のディスパッチ機会を待つ．ディスパッチャがこのタスクをディスパッチしようとする際に，タスクが確保要求をしているDB/DC のロックの空き状態を調べ，ロックが空いていればロックを確保させ，ディスパッチする．ロックが空いていなければ，タスクをレディ状態にしたまま次のディスパッチ機会を待たせる．

2.2 シミュレーションの方針

TCMP システムには，1章で述べたように，第1にプロセッサ1台あたりの実効速度の低下，第2にトランザクションの応答時間増加，第3に1トランザクションの処理に要する CPU オーバヘッドの増加の3つの性能低下要素がある．提案するシミュレーションモデルでは，ハードウェア動作に主に依存して発生するプロセッサ1台あたりの実効速度低下の評価は行わず，シミュレーションにあたりプロセッサの実行速度は，シミュレーション実行時のパラメタとして机上値を指定する．第2のトランザクションの応答時間の増加と第3の1トランザクションの処理に要する CPU オーバヘッドの増加の2つの要素についてはその主要な性能要因を詳細にモデル化する．こうすることによって TCMP のシステム性能の評価を行う．第2のトランザクションの応答時間の増加要因としては，オンラインシステムに複数個ある処理タスクの空き待ち時間や，OS のサスペンドタイプロック，DB/DC のロックの待ち時間の増加が主要なものである．要因としては他に IO 待ち，CPU 待ちがあるが，ここでは IO リソースが十分にあり IO 待ちはないものと仮定する．また CPU 利用率が高くなった場合，投入負荷を制限することを前提としているので，ここでは CPU 待ちは性能低下要因とは考えない．第3の1トランザクションの処理に要する CPU オーバヘッドの増加要因としては，タスクサーチや空間サーチ等のディスパッチオーバヘッド，OS のスピンタイプロックの競合による CPU のスピンオーバヘッド，OS のサスペンドタイプロック競合時の処理オーバヘッドの増加が主要なものである．提案モデルではこれらの要因を定量的に評価できるようにオンラインシステムの空間／タスク構成のモデル化，ディスパッチャの詳細なモデル化およびロック競合の詳細なモデル化を行うことを方針とした．

2.3 シミュレーションモデル

本節では，上述の方針にもとづいて作成したシミュレーションモデルの構成，各コンポーネントの働き，シミュレーションモデルの機能について述べる．モデルの作成にあたっては，実測環境でのトランザクショ

ンの命令トレースデータを解析して，タスク，アクション，割込み処理を抽出し，それらに1対1に対応するようにシミュレータ内にタスクモデル，アクションモデル，割込み処理モデルを持ち，命令トレースデータから，処理単価やロック保有ステップを求め，10〜100 ステップの精度でモデルに反映することを基本的な方針にした．ディスパッチャやロックマネージャ，SVC (Supervisor Call) 処理等，トランザクション処理のシミュレーションに必要な OS のカーネル部のモデルは実システムに基づいて作成した．OS カーネル部についても実行ステップ数を詳細にモデル化している．

(1) シミュレーションモデルの構成

以下，シミュレーションモデルについて説明する．まず，シミュレーションモデルは次のコンポーネントから構成されている．

(a) 環境モデル
- (i) CPU モデル
- (ii) トランザクション発生モデル
- (iii) IO モデル
- (iv) チャネルモデル

(b) OS モデル
- (i) ディスパッチャモデル
- (ii) ロックマネージャモデル
- (iii) 割込みハンドラモデル
- (iv) SVC 処理モデル
- (v) 割込み処理モデル
- (vi) アクションモデル

(c) アプリケーションモデル
- (i) システムタスクモデル
- (ii) 処理タスクモデル
- (iii) ユーザタスクモデル

(2) 各コンポーネントの働き

次に各コンポーネントの働きについて述べる．シミュレーションモデルの構造と動作を**図3**に示す．

(a) 環境モデル

(i) CPU モデル：CPU モデルは，チャネルモデルの動作を実行する1台の環境プロセッサと，シミュレーション開始時に指定する台数の命令プロセッサから構成される．環境プロセッサはチャネルモデルの動作の実行のほか，トランザクション発生部の動作の実行を行う．命令プロセッサはディスパッチャモデルやロックマネージャモデル，タスクモデル，アクションモデル，SVC 処理モデル，割込み処理モデル等を実行する．命令プロセッサは走行状態としてアイドル状態とビジー状態の2状態を持つ．また IO・外部割込みについて割込み可能状態と割込み禁止状態の2つの状態を持つ．命令プロセッサがタスクモデル等を実行中に割込みを受付けると，割込みに対応した割込み処理モデルに制御をうつす．割込み処理モデルの実行が終了すると，割込みハンドラモデルを起動し，命令プロセッサが次に何を実行すべきかを決定し，処理を継続していく．命令プロセッサは実行可能なタスクまたはアクションがなくなると，アイドル状態に入り，新たに割込みを受けるまでアイドルウェイトする．

(ii) トランザクション発生モデル：シミュレーション開始時に指定する値 (n) に従い，毎秒平均 n 個の IO 割込みを発生させ，チャネルモデルに割込みを掛ける．この IO 割込みはトランザクションの到着を表す割込みである．

(iii) IO モデル：タスクモデル等が SVC 処理モデル等を経由して発行した IO に対して，所定時間経過後，IO 割込みをチャネルモデルに対して返す．

(iv) チャネルモデル：トラン

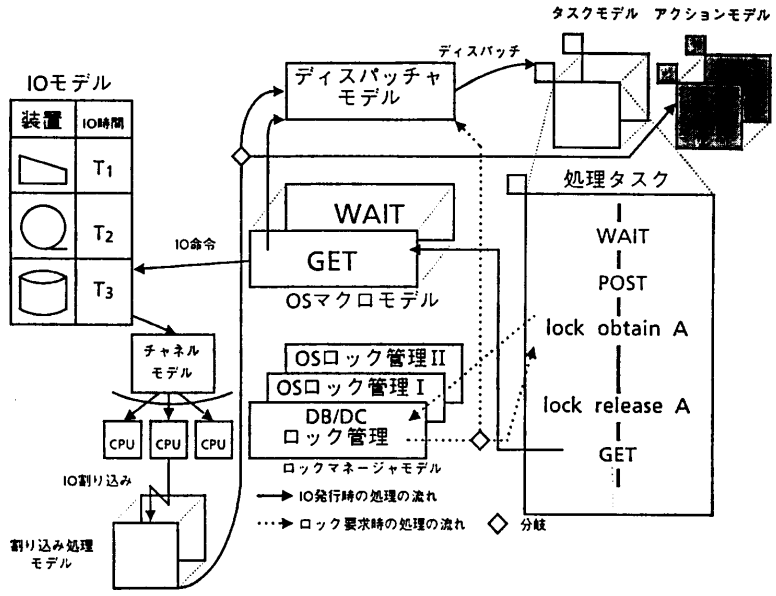

図3 シミュレーションモデルの構造と動作
Fig. 3 Structure and action of simulation model.

ザクション発生モデルと IO モデルから IO 割込みを受け取ると，各 CPU の状態を調べてどの CPU に割り込むかを決定し，選択した CPU に対してIO割込みを掛ける．

(b) OS モデル

(i) ディスパッチャモデル：まず空間キューのサーチを行い，レディタスクの存在する空間を見つける．そして，サーチした空間数に，空間にレディタスクがあるかどうかの判定に要するステップ数をかけ，前後の処理に要するステップ数を加えた時間だけ，スピンタイプロックであるディスパッチャロックを取得して CPU を消費する．これにより，プロセッサ台数や，オンラインシステムの空間数などのシステム構成が変化した場合に，空間サーチ時間や，ディスパッチャロックの競合によるスピン時間が変化する．次に見つけた空間のタスクキューロックの確保を試行する．タスクキューロックが確保できなかった場合，この空間のディスパッチはあきらめ，次の空間から空間サーチを再開する．タスクキューロックが確保できた場合，タスクキューサーチを行い，レディタスクを見つけディスパッチする．この際，サーチしたタスク数にタスクがディスパッチ可能かどうかの判定に要するステップ数をかけ，前後の処理に要するステップ数を加えた時間だけ，サスペンドタイプロックであるタスクキューロックを取得して CPU を消費する．これにより，プロセッサ台数や，システム空間の処理タスク数などのシステム構成が変化した場合に，タスクサーチ時間やタスクキューロックの保持時間が変化する．

(ii) ロックマネージャモデル：ロックマネージャモデルは，OS のスピンタイプロックマネージャモデル，OS のサスペンドタイプロックマネージャモデル，DB/DC のロックマネージャモデルの3つから成り立っている．3種のロックはすべて，ロックごとにシミュレータ内にロックワードを持ち，ロック確保要求時とロック解放要求時にロックワードのチェック，書き込みを行うことにより管理する．ロックマネージャモデルではロック管理のシミュレーションと同時に，ロックごとにロックの競合回数，保有時間等を計測する．

(iii) 割込みハンドラモデル：割込みハンドラモデルは，SVC 割込みハンドラモデルと IO・外部割込みハンドラモデルから成り立つ．SVC 割込みハンドラモデルは SVC の種類に応じて，SVC 処理終了後に，SVC 発行先かディスパッチャへ制御を移す．IO・外部割込みハンドラモデルは，割込み処理モデルの実行終了後に起動され，次の処理を行う．タスクが割込みを受けた場合にはディスパッチャへ制御を移し，アクションが割込みを受けた場合にはアクションの実行中断点へリターンする．

(iv) SVC 処理モデル：POST マクロや WAIT マクロなどトランザクション処理に現れるマクロを，実測環境で取得した実トランザクションの命令トレースデータから処理単価，ロック範囲を求め，SVC 処理モデルとしてモデル化している．モデルはロックの確保／解放要求の実行および CPU を消費する負荷部分と，タスクのウェイト処理等の OS マクロの処理をシミュレーションするロジック部分から成り立つ．

(v) 割込み処理モデル：SVC 処理モデルと同様に実トランザクションの命令トレースデータを元に，個々の IO 割込みごとに，割込み処理モデルを作成した．割込み処理モデルは，ロックの確保／解放の実行および CPU を消費する負荷部分と，アクションの生成やタスクの POST 処理等を行うロジック部分から成り立つ．

(vi) アクションモデル：アクションはトランザクション処理の中で IO 割込み処理や IO 発行処理の一部を実行するため等に使用される．オンライントランザクション処理時に発生するすべてのアクションをアクションモデルとしてモデル化した．アクションモデルはロックの確保／解放要求の実行および CPU を消費する負荷部分と，タスクの POST 処理等を行うロジック部分から成り立つ．

(c) アプリケーションモデル：アプリケーションモデルはシステムタスクモデル，処理タスクモデル，ユーザタスクモデルから成り立つ．それぞれ実測環境での実トランザクションの命令トレースデータを元に作成しており，POST マクロや WAIT マクロ，DB のアクセスマクロ等の SVC の発行，DB/DC ロックの確保／解放要求，CPU の消費部分等を時系列に並べたものである．

(3) シミュレーションモデルの機能

開発したシミュレーションモデルの機能を図 4 に示す．シミュレーションモデルは，プロセッサ台数，トランザクション発生率，DB/DC ユーザ空間数，MIPS 値を変化させてシミュレーションすることができる．出力データから，TCMP システムの性能予測，性能ネックの検知，性能低下要因の内訳分析が可能である．

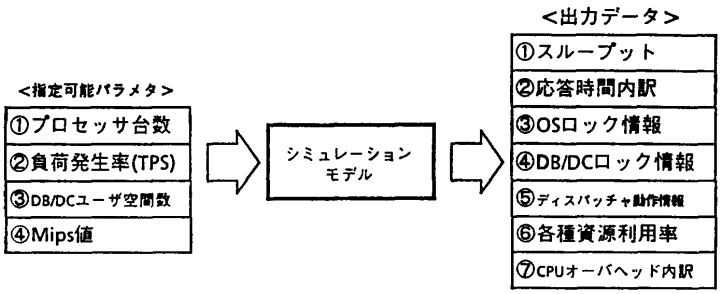

図 4 シミュレーションモデルの機能
Fig. 4 Function of simulation model.

2.4 シミュレーションモデルの特長と効果

提案したシミュレーションモデルの特長と効果は次のとおりである．(1)トランザクションの命令トレースデータを解析して，タスク，アクション，割込み処理を抽出し，それらに1対1に対応するようにシミュレータ内にタスクモデル，アクションモデル，割込み処理モデルを持ち，命令トレースデータから，処理単価やロック保有ステップを求め，10〜100 ステップの精度を保つようにモデル作成を行うとともに，ディスパッチャやロックマネージャ，SVC 処理などの OS のカーネル部は実システムをもとにモデル作成を行った．このようなモデル化を行うことにより，実システムに即した正確な性能評価，改善方式の効果の定量的評価が可能となった．(2)CPU モデルがサスペンドタイプの OS ロックや DB/DC ロックを保有中のタスクモデルやアクションモデルを実行中に割込みを受付けると，タスクモデルやアクションモデルの実行を中断して，割込み処理モデルを実行する．これにより，割込みによるロック保有時間の増加をシミュレートすることができる．またロック保有中のタスクモデルを実行中に割込みを受けた場合，割込み処理モデルの実行終了後ディスパッチャへ制御が移る．このとき，もともとロックを保有して走行していたタスクモデルより高いプライオリティのタスクモデルやアクションモデルがレディ状態になっていれば，それを実行するため，ロック保有中のタスクのディスパッチ遅れが生じる．これによりタスクのディスパッチ遅れによるロック保有時間の増加をシミュレートすることができる．ロック保有時間増加のメカニズムを**図5**に示す．(3)ディスパッチオーバヘッドは，単にディスパッチャが起動された時，一定時間 CPU を消費する単純なモデルではなく，空間キューおよびタスクキューのモデルをシミュレータ内にもち，タスクの状態としてレディ状態，走行状態，ウェイト状態，サスペンドロック待ち状態の4通りの状態をモデル化し，ディスパッチャのロジックに従って，空間およびタスクの

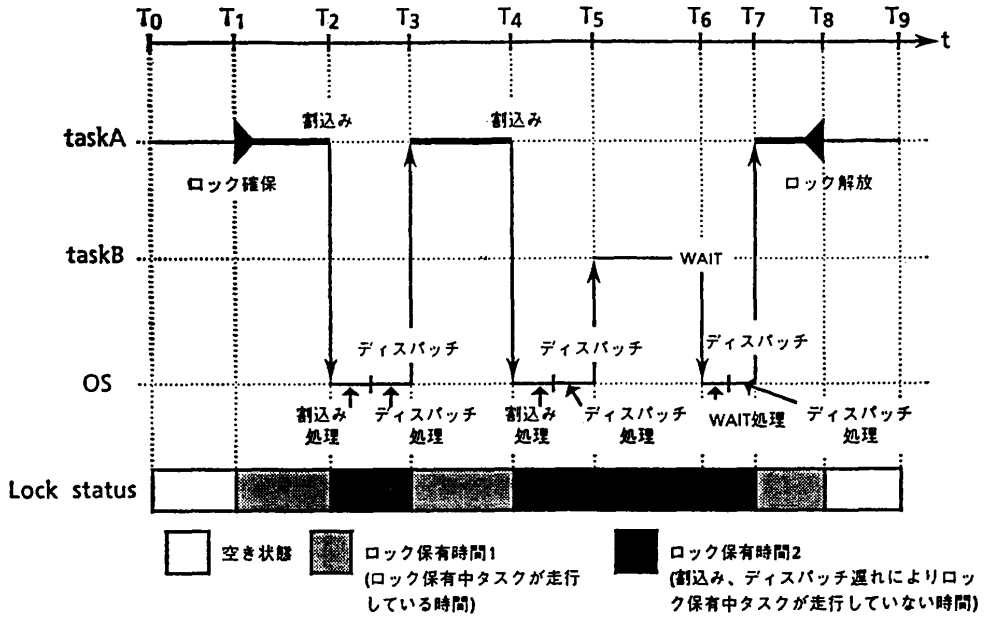

図 5 ロック保有時間増加のメカニズム
Fig. 5 Mechanism of increase of lock-holding time.

サーチを行い，空間／タスクのサーチ個数をカウントし，そのカウント値からキューサーチ時間を求め，その時間だけキューを排他制御するロック（ディスパッチャロックおよびタスクキューのロック）を保有してCPU を消費することにより，空間サーチオーバヘッドとタスクサーチオーバヘッドから成り立つディスパッチオーバヘッドと，ディスパッチャロックおよびタスクキューのロック競合オーバヘッドを定量的に評価できる．

3. シミュレーション例

2章で述べたシミュレーションモデルを TCMP システムの評価に適用した例について述べる．シミュレーションでは1トランザクション当りの I/O 回数が少なく，軽いトランザクションが多数集中するオンラインシステムを想定した．3.1節では，DB/DC ロックのロック保有時間が割込みやタスクのディスパッチ遅れにより増加しているシミュレーション結果を示し，3.2 節ではディスパッチオーバヘッドに密接に関わる空間の平均サーチ個数をシミュレーションにより求めた結果を示す．

3.1 ロック保有時間の評価

図6 はある1つの DB/DC ロックのロック保有時間の増加の様子を示している．ここでロック保有時間とはシミュレーションが単位時間進むあいだ，それぞれのロックが占有状態（いずれかのタスクがロックを保有し，他のタスクがロックを確保しえない状態）にあった時間を指す．ロック保有時間は，次の2つの時間に分類することができる．第1はロックを保有中のタスクが走行している時間である．第2はロックがいずれかのタスクに保有されているが，ロック保有中のタスクが割込み処理の実行や，ディスパッチ遅れにより，走行しておらず，ロックが無駄に保有されている時間である．図から割込みやディスパッチ遅れによりロックの保有時間が増加する現象がシミュレートされていることが分かる．したがって提案したシミュレーション手法によりロック競合率，ロック待ち時間についてロック保有時間が固定的なモデルに比べてより正確な評価が可能である．

3.2 空間サーチ個数の評価

図7 は，4台，8台，12台，16台構成の TCMP システムをシミュレーションし，ディスパッチャにおける平均の空間サーチ個数を求めたものである．シミュレーション実行時にはプロセッサ台数を大きくするに

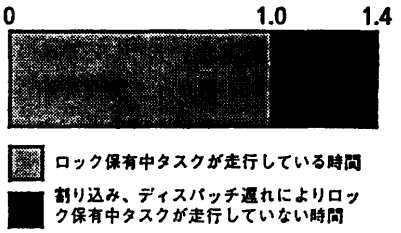

図 6 ロック保有時間の増加
Fig. 6 Increase of lock-holding time.

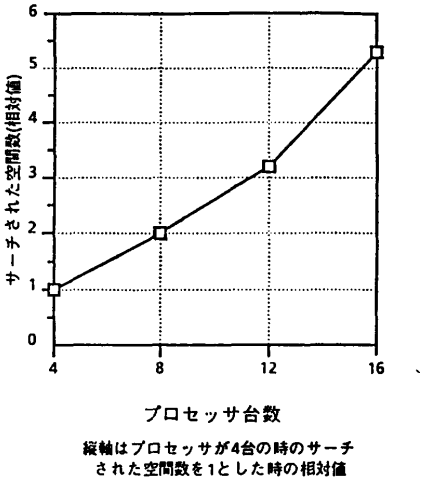

図 7 サーチされた空間数とプロセッサ台数の関係
Fig. 7 Relation between number of searched space and number of processors.

従い，投入負荷・空間数を増やし，CPU 利用率が 90 % 程度になるように，パラメタを調整している．プロセッサ数や空間数などのシステム規模の違いによる平均の空間サーチ個数の変化がシミュレーション可能であることがわかる．これにより，システム規模の変化を踏まえた定量的なディスパッチオーバヘッドの評価が可能である．

4. 精度検証

本章では，DB/DC ロックのロック競合度についてシミュレーションの精度検証を実測値との比較により行った結果を示す．**図8** および **図9** に 4 way 時に1個の DB/DC ロックのロック競合度を評価した結果を示す．グラフの横軸は図8，図9ともにプロセッサ1台当たりの CPU 利用率である．図8の縦軸はロック要求1回当たりの平均のロック待ち時間の相対値である．図9の縦軸はロック競合率（ロック要求回数と競

合回数の比）の相対値である．ケース1はロックの範囲が比較的大きいモデルである．ケース2はケース1のロック範囲を縮小し，性能改善を図ったモデルである．ロックの待ち時間，ロック競合率とも，実測とシミュレーションで良い一致が見られることが把握できた．

5. おわりに

密結合マルチプロセッサシステムにおいて，トランザクションの応答時間の増加と1トランザクションの処理に要するCPUオーバヘッドの増加という性能低下要素に着目し，処理タスクの空き待ち時間の増加，ロック競合による待ちの発生，タスクディスパッチ時のサーチオーバヘッドやロック競合によるオーバヘッド増加のメカニズムをシミュレーションすることにより，トランザクションの応答時間の増加と1トランザクション当たりのCPUオーバヘッドの増加を定量的に評価可能とするシミュレーションモデルを提案した．またシミュレーションモデルの構成，シミュレーション例，およびDB/DCのロックの競合度について実測値と比較を行った結果について述べた．DB/DCのロックの競合度はシミュレーション値と実測値で良く一致することが確かめられた．提案したシミュレーションモデルにより，TCMPシステムの定量的な性能評価が可能となった．

謝辞 本研究を推進するにあたりご協力をいただいた(株)日立製作所ソフトウェア開発本部，同情報システム開発本部の皆様に感謝いたします．

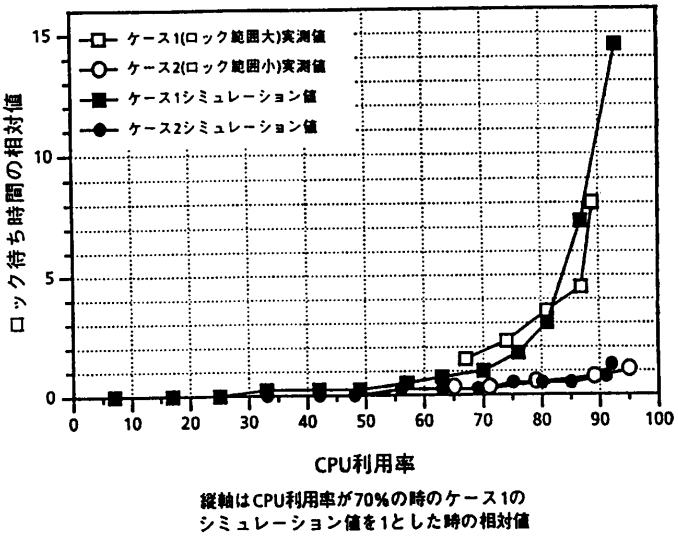

図8 シミュレーション値と実測値の比較
Fig. 8 Comparison of simulated value and measured value.

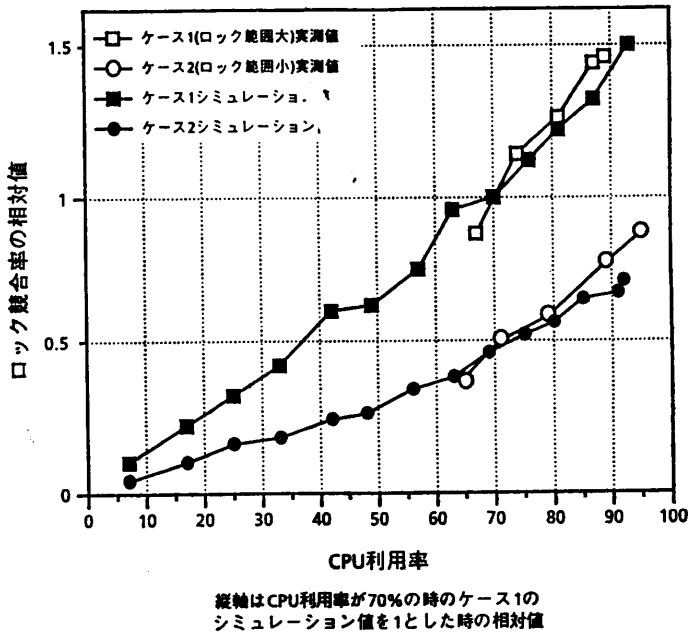

図9 シミュレーション値と実測値の比較
Fig. 9 Comparison of simulated value and measured value.

参考文献

1) 大特集 大型汎用コンピュータのすべて：日経コンピュータ，1990. 10.15号 (1990).
2) 松岡，堀川，難波：マルチプロセッサシステムの評価技法と評価システム，情報処理学会研究報告，Vol. 89, No. 17, 89-OS-42 (1989).
3) 長坂，黒川，栗山，和田：密結合マルチプロセッサ記憶階層性能評価手法，情報処理学会研究報告，Vol. 90, No. 7, 90-ARC-80 (1990).
4) Vrsalovic, D. F., Siewiorek, D. P., Segall, Z. Z. and Gehringer, E. F.: Performance Prediction and Calibration for a Class of Multiprocessors, *IEEE Trans. Comput.*, Vol. 37, No. 11, pp. 1353-1365 (1988).
5) 松永，福村：バス結合型マルチプロセッサ方式の性能評価，電子情報通信学会論文誌 D-I, Vol. J73-D-I, No. 9, pp. 737-745 (1990).
6) 岩嵜，高本，吉住：密結合マルチプロセッサにおける排他制御処理オーバヘッドの解析的一評価

手法，情報処理学会論文誌，Vol. 31, No. 11, pp. 1627-1635 (1990).
7) 三上，久保，高橋，有福，北浦：コンピュータ・システム性能評価シミュレータ PACSS，情報処理，Vol. 12, No. 1, pp. 14-25 (1971).
8) 新井，吉澤：性能評価シミュレータ PRIME 3 の開発，第 21 回情報処理学会全国大会論文集，pp. 445-446 (1980).
9) 三上，亀田：コンピュータと情報システムにおけるシミュレーション技術，電子通信学会誌，Vol. 60, No. 7, pp. 761-769 (1977).
10) 根岸，藤井，住田，米田，佐藤：オンラインシステムを対象とした高多重プロセッサ高性能制御方式の提案，第 42 回情報処理学会全国大会論文集，pp. 4-21, 22 (1991).
11) 藤井，根岸，佐藤，吉澤：密結合型マルチプロセッサシステム性能評価シミュレータ PMOS の開発，第 40 回情報処理学会全国大会論文集，pp. 752-753 (1990).
12) 藤井，根岸，住田，米田，福多：シミュレータによるマルチプロセッサシステムの性能評価，第 42 回情報処理学会全国大会論文集，pp. 4-69, 70 (1991).

(平成 3 年 2 月 27 日受付)
(平成 3 年 10 月 3 日採録)

藤井 哲彦（正会員）

1982 年東京大学理学部数学科卒業．1985 年同大学院理学系数学科修士課程修了．同年(株)日立製作所入社．以来同社システム開発研究所において，マルチプロセッサシステム，オンラインシステム，性能評価，ファイルシステムの研究に従事．電子情報通信学会会員．

根岸 和義（正会員）

1973 年東京工業大学電子物理工学科卒業．1975 年同大学院理工学専攻科修士課程卒業．同年(株)日立製作所入社．以来同社システム開発研究所において分散データベース，オンラインシステム，マルチプロセッサシステムの研究に従事．現在，同研究所第 6 部主任研究員．電子情報通信学会，ACM，IEEE CS 各会員．

米田 茂（正会員）

昭和 22 年生．昭和 45 年名古屋工業大学工学部計測工学科卒業．昭和 45 年日立製作所情報システム研究所入所．現在，同社システム開発研究所第 6 部長．入社以来，データベースコントロールシステムの開発，性能評価，高信頼化機能，分散機能，エンドユーザインタフェース，等 DB/DC 関連の研究開発に従事．