

並列計算機の解析モデル — 共有ブロックの予測 —

城 和貴[†] 福田 晃[‡]

[†] 奈良先端科学技術大学院大学情報科学科
 ◇ 九州大学工学部情報工学科

Abstract

キャッシュを有した共有メモリ型並列計算機の解析モデルを構築する際に、キャッシュ・コヒーレンス制御のモデル化は重要かつ構築困難なところである。これまでに提案されているモデルには、データを共有ブロックとプライベート・ブロックに分離することにより、共有ブロック、すなわちコヒーレンス制御のモデル化を試みているものがある。このモデルは簡潔である反面、プライベート・ブロックへのミス・ヒットが共有ブロックに影響を与えない、という前提条件が必要であった。しかしながら、並列計算機における実際のアプリケーションを考えた場合、プライベート・ブロックへのミス・ヒットが無視できない規模のプログラムも多い。そこで本稿では、プライベート・ブロックへのミス・ヒットが共有ブロックに影響を与えることを考慮した、並列計算機の解析モデルを提案する。マルコフ連鎖を用いて構築された本解析モデルでは、プライベート・ブロックに対するキャッシュ・ヒット率を与えることで、共有ブロックがどのような状態に分布するかを予測する。

An Analytic Model for a Multiprocessor Computer — Prediction of a Shared Block Behavior —

Kazuki JOE Akira FUKUDA

Graduate School of Information Science
Advanced Institute of Science and Technology, Nara
8916-5, Takayama, Ikoma, Nara 630-01 Japan

E-mail: {kazuki-j, fukuda}@is.aist-nara.ac.jp

Abstract

Developing an analytic model for a cache based shared memory parallel computer, a part of modelling for a cache coherence control is one of the most important and difficult stuff. Some previous model attempted to focus on shared blocks from the view point of cache coherence mechanism under the assumption of infinite cache size. Although the model is simple, there is the unavoidable assumption that the miss hit of private block does not affect the shared block behavior. Considering a large-scaled number crunching, there are many applications which can not ignore the affection of such a miss hit. In this paper, we propose a new analytic model of a parallel computer with finite cache size where the miss hit of private blocks affects shared block behavior. Using Markov process, our model predicts the state of a shared block when a cache hit ratio of private block is given.

1 はじめに

並列計算機アーキテクチャは、十数年前には、単純に接続された複数のプロセッサとメモリ・モジュールの形態が主流であった。その後、キャッシュを持ちコヒーレンス制御を行なうような複雑なもの、クラスタ化による大規模なもの、それらの両方を許すもの、と数多くの研究がなされ、今日に至っている。

このように次々と提案されるアーキテクチャに、適切な性能評価もしくは性能予測を与えることは、アーキテクチャの提案自体に優るとも劣らない重要な研究課題であろう。通常、並列計算機アーキテクチャ研究のために必要とされる性能評価は、シミュレーションや解析モデルで行なわれる。この時、評価対象が何であるかによって、その評価をもっとも効果的に行なえる、シミュレーション・モデルや解析モデルが選ばれる。例えば、クロスバー・ネットワーク等で結合された単純な並列計算機の理論的なメモリ・バンド幅を求めるだけならば、確率と組合せによる解析モデル [2] で瞬時に答が得られる。一つの並列プログラムを全プロセッサで実行させた時の、プロセッサ利用率やネットワーク使用率を求めるには、与えられたプログラム・モデルが単純なものであれば、確率過程等を利用した解析モデル [9][10] や、簡略化されたシミュレーション等 [5] が適当である。そのプログラムが非均等な同期命令を含むのであれば、時間ベトリ・ネットのような高性能の解析モデル [8] や、詳細なシミュレーションに頼らざるをえないであろう。

我々は、キャッシュのない共有メモリ型並列計算機に対する T.N. Mudge の提案した SMI モデル (Semi-Markov Memory Interference Model) [10] を拡張し、キャッシュを有しコヒーレンス制御を行なう並列計算機に対する解析モデルの提案を行ってきた [14][12][13]。我々の一連の解析モデルの目的は、並列計算機における、メモリ・アクセスおよびキャッシュ・コヒーレンス制御のリクエストの競合状態の解析であった。一方、キャッシュ付き並列計算機の解析モデルとしては、M. Dubois の提案するアクセス・バースト・モデルが知られている [7]。この研究の目的は、プログラム・モデルに応じた理論的なキャッシュ・ミス率を解析することであった。しかしながら、Dubois のモデルでは、キャッシュは無限にあるという仮定のもとでの解析モデルであり、非共有ブロックは完全にキャッシュ・ヒットし、コヒーレンス制御やリプレースに起因する、キャッシュ・ミス率の解析は行なえない。

最近の LSI 技術の進歩はめざましく、Dubois らの主張するように、キャッシュが無限に使えという状況は、年を追うごとに正当性をおびてくる。しかしながら、これは既存サイズの問題を新しい計算機で行なう場合に言えることであって、計算

機の性能が大幅に向上すれば、データ・サイズの拡大や問題の複雑化も、それに応じて求められるのではないだろうか。そこで、我々は、技術革新により計算機が高性能かつ大規模化しても、それと同時に要求されるデータ・セットも巨大化するため、キャッシュ・ミスに関わる性能低下は、依然として性能評価に本質的なものである、という着眼点から出発する。

本稿では、Dubois のモデルを拡張・修正し、有限長のキャッシュにおいて、共有データが非共有データのヒット率のどのような影響を受けるかを考察する。モデル化にあたっては、対象アーキテクチャとしてセット・アソシアティブ方式のキャッシュを持った共有メモリ型並列計算機を想定し、セット・アソシアティブ・マッピングの特性を利用した、共有/プライベート・ブロック分離型および混在型の概念を導入し、後者に対する解析モデルをマルコフ連鎖を用いて提案する。第 2 章では本研究の出発点となった Michel Dubois のキャッシュに対する解析モデルについての概略を述べ、第 3 章で実際のモデル構築法について説明し、第 4 章で簡単な評価実験結果について報告する。

2 Michel Dubois のモデル

2.1 共有ブロックとプライベート・ブロック

1982年、Michel Dubois らは並列計算機における、キャッシュ・コヒーレンスの解析モデルを提案している [6]。提案されたモデルでは、具体的なプロトコルが採用されているわけではないが、キャッシュ制御のワークロードを与え、マルコフ連鎖を用いて、コヒーレンス制御を行なう場合と、行なわない場合について、キャッシュ・サイズがヒット率に与える影響の定式化を行なっている。この研究において注目すべきところは、並列プログラムにおいて、データ領域を共有ブロックと、プライベート・ブロックに分け、キャッシュの動きを、それぞれのブロックに応じた形でモデル化したことである。図 1 で示すように、データ・アクセスは確率 q_s で共有ブロックに、 $1 - q_s$ でプライベート・ブロックに対してなされる。共有ブロックでは、特定の分布に従ったアクセスをするのに対し、プライベート・ブロックでは、LRU スタックに従った形でアクセスがなされる。共有/プライベート・ブロックの概念は、以後のキャッシュ付き並列計算機 (バス結合に限らない) に対する解析モデルに、大きな影響を与えている。コヒーレンス・チェックを行なわない場合のモデル化では、キャッシュ・サイズと、キャッシュ・ヒット率に関する定式化を行ない、プロセッサからのリクエストが起きるたびに状態を変えるマルコフ連鎖で、全体が

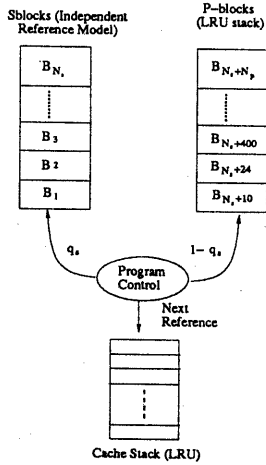


図 1: Dubois による共有ブロックとプライベート・ブロックの概念

簡単に表されている。ここでの目的は、共有ブロックへのアクセスが、全体のキャッシュ・ヒット率にどのような影響を与えるか、を分析することである。コヒーレンス・チェックを行なう場合のモデル化では、特定のプロセッサに着目し、連続して起こる 2 回のリクエストの間に、他のプロセッサはそれぞれ 1 回ずつリクエストを起こすという仮定を与え、インバリデーションや、それに伴うリプレースを考慮にいたれた定式化について報告している。図 2 は、コヒーレンス・チェックを考慮に入れたモデルを示す状態遷移図である。同モデルは、

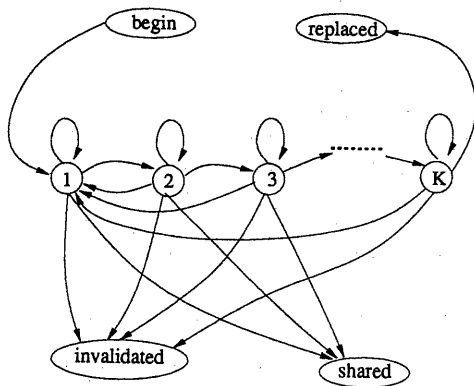


図 2: Dubois によるマルコフ連鎖を用いたコヒーレンス・チェック付き並列計算機のモデル化

仮想時間¹で表されているため、プロセッサの利用率や、リクエストの待ち時間等の解析が不可能で

¹これに対して、例えばネットワークのサイクルごとに状態を変えるようなモデルは、実時間で表されていると言う。

あり、かつ、各プロセッサのリクエストの出し方に制限が大きいため、キャッシュ付き並列計算機を正しくモデル化しているかどうかは疑問が残る。しかしながら、キャッシュ・コヒーレンス問題に対する研究がホットな時期のモデルであることを考慮すると、Dubois の研究は特筆に値するものであろう。

2.2 プログラム・ビヘイビア・モデル

1987年に Dubois は、プログラム・ビヘイビア・モデルに基づく並列計算機の解析モデルを提案している [4]。これは文献 [6] の共有/プライベート・ブロックの概念を、プログラム・ビヘイビア・モデルに適用し、キャッシュ・ブロックの挙動を予測して、インバリデーションがキャッシュ・ヒット率にどのような影響を与えるかをあらわしたものである。プログラム・ビヘイビア・モデルとしては、各プロセッサが独立して共有ブロックにアクセスを行なう Independent Reference Model (IRM)、各プロセッサがクリティカル・セクションを実行中の時のみ共有ブロックにアクセスを行なう Independent Reference Model with Critical Section (IRMCS)、同じくセミ・クリティカル・セクションを実行中の時のみ共有ブロックにアクセスを行なう Independent Reference Model with Semi-Critical Section (IRMSCS) の 3 種類が提案されている。実際のモデル化は、キャッシュは無限にあるという仮定のもとで、共有ブロックの挙動をマルコフ連鎖を用いて構築している。

2.3 アクセス・バースト・モデル

1988年に Dubois らは、具体的な並列プログラムに対する解析モデルの提案を行なっている [7]。これは文献 [4] のモデルの、具体的なプログラムへの対応を許すもので、アクセス・バーストという新しい概念を導入している。ある並列プログラムが無限長のプライベート・キャッシュを持つ並列計算機で安定稼働している時、特定の共有ブロックは図 3 で示すように、ダーティ (1_{RW}) もしくは k 個のプロセッサによって共有 (k_{RO}) されている状態にある。状態遷移は、プログラムにおいて、クリティカル・セクション (CS) もしくはセミ・クリティカル・セクション (SCS) を脱出する瞬間に起きるものとする。あるプロセッサが CS や SCS を実行中の時、そのプロセッサはアクセス・バーストを行なっていると考える。このモデルにより、与えられたプログラムのアクセス・バースト長 (CS や SCS におけるリクエストの数)、アクセス・バーストにおけるリード・ライトの順番等の情報を、パラメータとして与えてやると、共有ブロックのミス率およびコヒーレンス

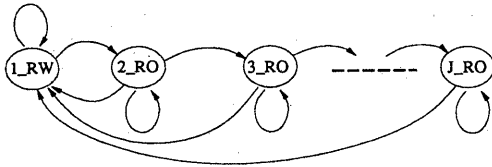


図 3: Dubois のアクセス・バースト・モデルによる共有ブロックの状態遷移

制御のためのオーバーヘッドを計算することができる。Dubois らはさらに、実際の並列プログラム (S. O. R., FFT、クイック・ソート) に対してこの解析モデルを適用し、文献 [5] で報告しているシミュレーション結果との比較を行ない、提案した解析モデルの妥当性を示している。Dubois の解析モデルは、文献 [6] 同様、仮想時間により推移するマルコフ連鎖を用いているため、プロセッサの待ち状態や利用率を求めることができず、無限長のキャッシュを想定しているためプライベート・ブロックがミス・ヒットしたために引き起こされるリプレースの対象として、共有ブロックが選ばれる様子がモデル化されない。しかしながら、実際の並列プログラムの性能予測を可能とした初めての解析モデルとして、高い評価に値する研究と見えよう。

Jin-Chin Wang らは、1990年に Dubois のアクセス・バースト・モデルを Write-Once、Synapse、Illinois、Berkeley 等、さまざまなコヒーレンス・プロトコルに拡張し、SOR の並列プログラムに適用した結果、Archibald のシミュレーション結果と同様の傾向が得られたことを報告している [11]。

3 共有ブロック状態予測モデル

3.1 表記法

- P プロセッサ (キャッシュ) の個数
- h プライベート・ブロックに対するキャッシュ・ヒット率
- H 共有ブロックに対するキャッシュ・ヒット率
- r リード・リクエスト率
- u 共有ブロック・アクセス率
- N_p プライベート・ブロック数
- N_s 共有ブロック数
- E キャッシュ・エントリ数
- W キャッシュのウェイト数
- E_p キャッシュ中のプライベート・ブロックの数
- E_s キャッシュ中の共有ブロックの数
- p_i プライベート・ブロックに対する確率分布
- s_i 共有ブロックに対する確率分布

π_k 共有ブロックの状態を表す

R_p リプレース対象に特定の共有ブロックが選ばれる確率

3.2 モデルの仮定

有限長キャッシュを有する共有メモリ型並列計算機の、共有ブロックの動作予測を行なうモデルを構築するにあたって、以下のような仮定を行なう。

1. 各プロセッサは共有ブロック、プライベート・ブロックに対して、それぞれ与えられた分布関数に従ったアクセスを行なう。
2. 全てのプロセッサからのリクエストが全て独立であるような並列プログラムが稼働中であるとす。言い換えれば特殊な同期操作は本モデルでは記述出来ないことになる。
3. プライベート・ブロックに関しては、定常状態を仮定する。すなわち、共有ブロックへのアクセスが始まる前に、プライベート・ブロックの初期アクセスに伴うミス・ヒットは終了しているものとする。
4. キャッシュ・コヒーレンス・プロトコルとしては、Synapse を用いる。ただし、他の任意のプロトコルに関して、本モデルは容易に適用可能である。

3.3 モデル化の方針

前章で述べたように、Dubois の共有ブロックのモデルは、無限長のキャッシュを仮定することで、プライベート・ブロックおよび共有ブロックへのアクセスを分離してモデル化することに成功している。本稿では、有限長のキャッシュにおいて、常にある程度のミス・ヒットがプライベート・ブロックで発生しており、それが共有ブロックの状態にどのような影響を与えるかを調べることに主目的である。このため、Dubois のアクセス・バースト・モデルのように、仮想時間によって遷移する状態を考えたのでは、モデル化が困難である。本モデルでは、プライベート・ブロックへのアクセスを考慮するため、実時間によって遷移する状態を考える。

次節で述べるように、本モデルでは現在広く利用されているセット・アソシアティブ・マッピングによるキャッシュを考察するが、共有/プライベート・ブロックの概念をセット・アソシアティブ方式のキャッシュに適用した場合、共有/プライベート・ブロック分離型および同混在型の 2 種類に大別される。

共有/プライベート・ブロック分離型とは、図 4 で示すように、共有ブロックとプライベート・ブ

ロックを、キャッシュの異なるエントリーにマッピングされるように、メモリに配置したものである。この方式では、プライベート・ブロックに対するリクエストは、共有ブロックに対して本質的な影響を与えない。すなわち、プライベート・ブロックに対するミス・ヒットが共有ブロックをリプレースすることはない。従って、Duboisのアクセス・バースト・モデルの仮定を満たしており、同モデルのワークロードとして適用可能である。さらに、最近の研究では、キャッシュを共有/プライベートの部分に物理的に分割し、プライベート・ブロックへのアクセスが共有ブロックの状態に影響を与えないようなアーキテクチャが提案されている [3][1]。ここで提案する分離型の具体的なマッピング方式が定めれば、このようなキャッシュを分離した新しいアーキテクチャの効力は薄れ、既存のセット・アソシアティブ・キャッシュの潜在能力を十分に引き出せるものと推定される。

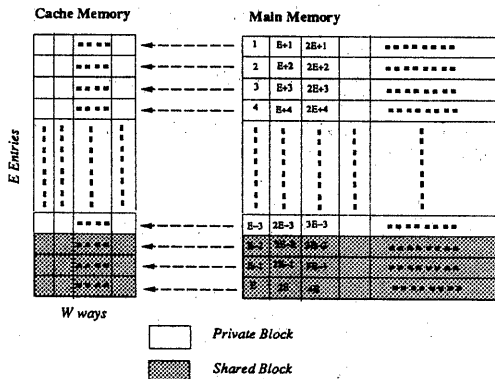


図4: 共有/プライベート・ブロック分離型のマッピング

共有/プライベート・ブロック混在型とは、図5で示すように、共有ブロックとプライベート・ブロックが一樣分散した状態でマッピングされたメモリ配置である。これは各ブロックのメモリへの配置を考慮しない、現在のシステムを代表するものと考えられる。この方式では、プライベート・ブロックに対するリクエストが、共有ブロックの状態に対して本質的な影響を与え得る。例えば、 i 番目のキャッシュ・エントリにおいて、最も古いアクセスが共有ブロックであった場合、プライベート・ブロックへのミス・ヒットが発生すると、リプレース対象として、その共有ブロックが選ばれる。通常、共有ブロックはプライベート・ブロックより、かなり少なく、アクセス率も極めて小さい。このため、同期操作等、特殊な命令列稼働中以外は、プライベート・ブロックのミス・ヒットが、かなりの確率で共有ブロックのリプレースを引き起こす

ものと考えられる。

本解析モデル構築の方針は、共有/プライベート・ブロック混在型のシステムに対して、実時間によって状態が遷移するモデルを考えることにより、プライベート・ブロックへの実際のアクセスを反映させた、共有ブロックの状態予測を行なうことである。

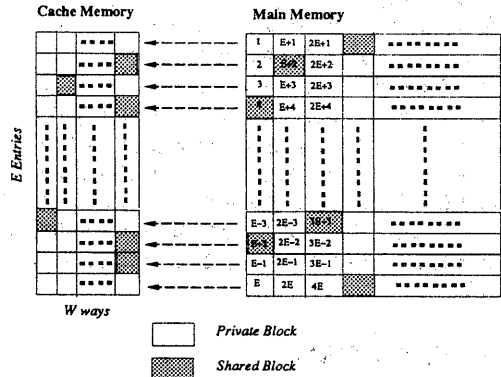


図5: 共有/プライベート・ブロック混在型のマッピング

3.4 対象とするアーキテクチャとそのワークロード

本稿で構築する解析モデルの対象とする並列計算機アーキテクチャは、メモリ共有型で、プロセッサごとにキャッシュを持ち、キャッシュ・コヒーレンス制御をハードウェアによって行なうものである。ネットワークやメモリ・モジュールの数はモデルに影響を与えない。ただし、キャッシュのマッピング方式は、ウェイ数 W のセット・アソシアティブ方式を考える。

今、上記アーキテクチャの並列計算機が、ある並列プログラムを実行しているとする。共有ブロックの状態予測は、どの共有ブロックもアクセスされていない状態から始まる。ただし、仮定により、プライベート・ブロックについては、既に安定稼働していると考えられる。プログラムは N_p 個のプライベート・ブロック、 N_s 個の共有ブロックを持ち、各キャッシュのエントリ数は E 、ウェイ数は W とする。ある時点におけるプライベート/共有ブロックのキャッシュを占める数を、それぞれ E_p 、 E_s とする。ただし、 $E \cdot W = E_p + E_s$ 。 P 個のプロセッサは、それぞれ各サイクルごとに f の割合でリクエストを発行する。このリクエストは、 u の確率で共有ブロックに対して発行される。従って、プライベート・ブロックに対する確率は、 $1-u$ である。リクエスト先がプライベート・ブロックの場合

合、そのアクセス・パターンは、LRUスタックの先頭からの距離 i に対して p_i の確率分布で表されるとする。ただし、 p_i は次のように定められる [6]。

$$p_i = \begin{cases} \frac{G(N_p)}{b} \left(\frac{1}{i-1} - \frac{1}{i} \right) & \text{for } i \neq 1 \\ G(N_p) = \left(1 - \frac{1}{b} \right) & \text{for } i = 1 \end{cases}$$

共有ブロックに対するアクセス・パターンも、同様にある確率分布 s_i で表されるが、本稿では簡単のため、一様分布を仮定する。従って、 $s_i = \frac{1}{N_s}$ である。プロセッサからのアクセスがプライベート・ブロックに対してなされた場合、キャッシュ・ヒットする確率を h とする。ミス・ヒットした場合、リプレースが発生する。本モデルは共有/プライベート・ブロック混在型を仮定しているので、この時のリプレース対象として、共有ブロックが選ばれる場合がある。この確率を R_p とする。厳密な R_p を求めることは難しいが、本モデルが混在型を前提としていることから、ここでは、 $R_p = \frac{1}{P}$ で近似する。

3.5 モデルの構築

本稿で構築されるモデルは、共有ブロックの状態をマルコフ連鎖を用いて表される。ある共有ブロック x の状態は、

$$\{\pi_{-1}, \pi_0, \pi_1, \dots, \pi_P\}$$

で表される。ただし、 π_{-1} はそのブロックが *Dirty* である状態、 π_i はそのブロックが i 個のプロセッサ (キャッシュ) によって共有されている状態を表す。(従って、 π_0 はキャッシングされていない。)

まず、 π_{-1} からは、その時のオーナー以外のキャッシュで、共有ブロック x に対するリードがあった時、 π_1 に、オーナー・キャッシュにおいてミス・ヒットが起り、リプレース対象に共有ブロック x が選ばれた時、 π_0 に、それ以外は自分自身に遷移する。 π_0 からは、共有ブロック x に対するリードが起れば π_1 に、ライトの場合は π_{-1} に、それ以外は自分自身に遷移する。 i 個のキャッシュによって共有ブロック x が共有されている状態 π_i からは、それ以外のキャッシュから共有ブロック x に対してリードがあった場合には、 $i+1$ 個のキャッシュによって共有され、また、 i 個のキャッシュのどれかでリプレース対象に共有ブロック x が選ばれた時には、 $i-1$ 個のキャッシュによって共有されることになる。さらに、共有ブロック x に対するライト要求があった場合には、状態 π_{-1} に遷移する。図 6 は、これらの状態遷移を示している。

特定の共有ブロックに対し、あるサイクルにおいて少なくとも 1 つのプロセッサが、ライト・リクエストを発行する確率を Ψ_w 、同じく特定の共有ブロックに対し、 k 個のプロセッサのうち、少なくとも 1 つのプロセッサがリード・リクエストを発行する確率を $\Psi_r(k)$ 、特定のキャッシュで、あるサイクルにおいてミス・ヒットが発生し、その結果、

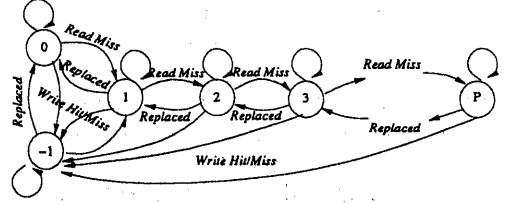


図 6: 共有ブロックの状態遷移

リプレースによって (キャッシングされていた) 特定の共有ブロックがリプレース対象に選ばれる確率を Φ とする。この時、 Ψ_w 、 $\Psi_r(k)$ 、 Φ は次のように表される。

$$\begin{aligned} \Psi_w &= 1 - (1 - u s_i f (1 - \tau))^P \\ \Psi_r(k) &= 1 - (1 - u s_i f \tau)^k \\ \Phi &= \left\{ (1 - u)(1 - h)f + \frac{P-1}{P} \Psi_r(P-1) \pi_{-1} \right. \\ &\quad \left. + \sum_{i=0}^{P-1} \frac{P-i}{P} \Psi_r(P-i) \pi_i \right\} R_p \end{aligned}$$

この時、特定の共有ブロックの各状態における状態方程式は、次で与えられる。

$$\begin{aligned} \pi_{-1}^{(j+1)} &= \sum_{i=0}^P \Psi_w \pi_i^{(j)} + (1 - \alpha(1) - \beta(1)) \pi_{-1}^{(j)} \\ \pi_0^{(j+1)} &= \alpha(1) (\pi_{-1}^{(j)} + \pi_1^{(j)}) + (1 - \beta(0) - \Psi_w) \pi_0^{(j)} \\ \pi_1^{(j+1)} &= \beta(0) \pi_0^{(j)} + \beta(1) \pi_{-1}^{(j)} + \alpha(2) \pi_2^{(j)} + (1 - \gamma(1)) \pi_1^{(j)} \\ \pi_k^{(j+1)} &= \beta(k-1) \pi_{k-1}^{(j)} + \alpha(k+1) \pi_{k+1}^{(j)} \\ &\quad + (1 - \gamma(k)) \pi_k^{(j)}, (k=2, 3, \dots, P-1) \\ \pi_k^{(P)} &= \beta(P-1) \pi_{P-1}^{(j)} + (1 - \alpha(P) - \Psi_w) \pi_k^{(j)} \end{aligned}$$

ただし、 $\{\pi_k^{(j)}\}$ は、状態 k の、 j サイクル後の確率ベクトルを表す。また、 $\alpha(k)$ 、 $\beta(k)$ 、 $\gamma(k)$ は次で与えられる。

$$\begin{aligned} \alpha(k) &= \frac{k}{P} \Phi \\ \beta(k) &= \frac{P-k}{P} \Psi_r(P-i) \\ \gamma(k) &= \alpha(k) + \beta(k) + \Psi_w \end{aligned}$$

4 評価

本モデルを利用した評価実験を行なった。ベース・パラメータとして、プロセッサ数 4、共有ブロック率 0.1、リクエスト率 1.0、リード・リクエスト率 0.7、プライベート・ブロックのキャッシュ・ヒット率 0.95、キャッシュ・エントリ数 256、ウェイ数 4、共有ブロック数 128、遷移回数 2,000 を考

える。評価対象は、各状態の分布である。

図7は、初期状態（全ての共有ブロックはキャッシングされていない状態）から、2,000サイクルの遷移を行ない、各状態がどのように収束していくかを示している。まず、Clean-Exclusiveの状態が多くなり、次第に各共有状態が増えていく様子が見えてくる。

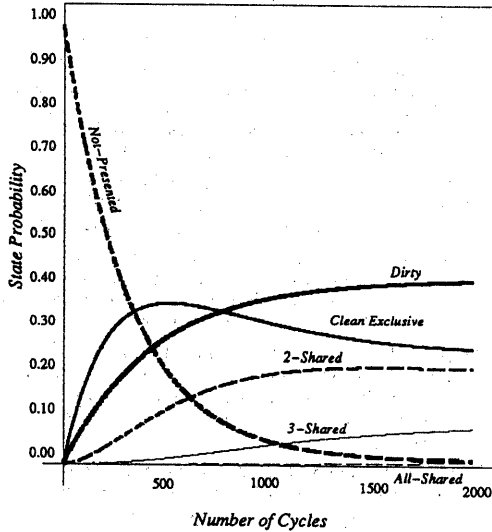


図7: 遷移回数と各状態の分布

図8は、プライベート・ブロックのキャッシュ・ヒット率を、図9は、リード・リクエスト率を、それぞれ変化させた時の、各状態の分布を示す。プライベート・ブロックのキャッシュ・ヒット率が増大すると、共有ブロックのClean-Exclusive状態を減少させ、Dirty状態や他の共有状態を増大させることが観察される。これは、プライベート・ブロックのミス・ヒットに引き起こされるリプレースメントが減少したためと推測される。

図10は、共有ブロック・アクセス率を変化させた時の、各状態の分布を示す。共有ブロック・アクセス率が増大すると、Dirty状態が急激に減少し、複数共有の状態が増加している。これは実際のシステムを考えた場合、ライト・アクセスに引き続いて、すぐにインバリデーションが発行され、システムの負荷が高くなっていることが推測される。

5 結論

セット・アソシアティブ方式のキャッシュを持つ並列計算機に対する、共有/プライベート・ブロック分離型および混在型の概念を示し、混在型に対する共有ブロックの状態予測を行なう解析モデル

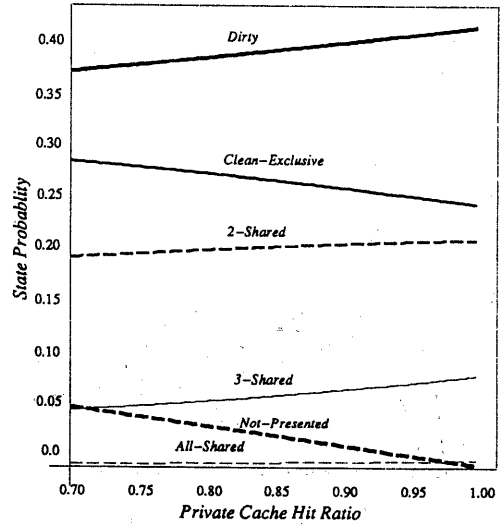


図8: プライベート・ブロック・キャッシュ・ヒット率と各状態の分布

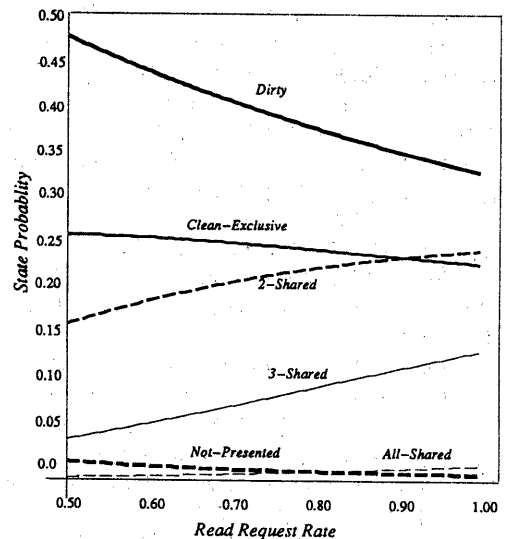


図9: リード・リクエスト率と各状態の分布

を、マルコフ連鎖を利用して表した。並列計算機におけるキャッシュは、セット・アソシアティブ方式を用いることが多いが、キャッシュ・ブロックのリプレースメントは各キャッシュ・エントリごとに行なわれるため、キャッシュ・エントリに対して一様に分散した形の共有ブロック配置となる可能性が大きい。本モデルは、このような共有/プライベート混在型マッピングに対して、プライベート・

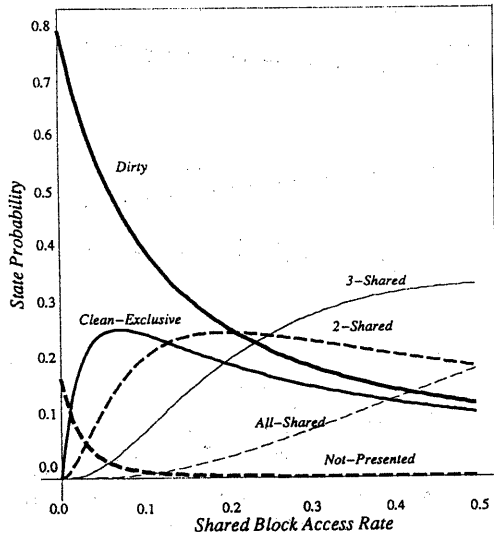


図 10: 共有ブロック・アクセス率と各状態の分布

ブロックに対するミス・ヒットが、共有ブロックの状態にどのような影響を与えるかを解析する。

本解析モデルを用いて、共有ブロックの状態予測を行なう評価実験を行なったところ、プライベート・ブロックのキャッシュ・ヒット率が、共有ブロックの状態に影響をあたえることが判明した。

一方、共有/プライベート分離型は、既存の解析モデルに対する無限長のキャッシュという仮定を与えなくても利用できることが判明し、その具体的なマッピング方式が与えられれば新しいキャッシュ方式を利用せずとも、既存のセット・アソシアティブ・キャッシュの能力を引き出せる可能性があることを示した。

本モデルにより、共有/プライベート・ブロック分散型マッピングにおける、共有ブロックのキャッシュ・ヒット率を容易に求めることができる。これは、先に報告した我々の並列計算機解析モデル [13] に対する入力パラメータとして利用できる。今後の研究課題は、これらの解析モデルを組み合わせ、ハイブリッドなモデルを構築し、詳細な評価実験を行なうことである。また、共有/プライベート分離型混合型に対する定量的な評価も、今後の重要な研究課題である。

参考文献

[1] M.S. Algudady, C.R. Das, and M.J. Thazhuthaveetil. A write update cache coherence protocol for min-based multiprocessors with accessibility-based split caches. In *Supercomputing 90*, pages 544-553, 1990.

[2] Laxmi N. Bhuyan. A combinatorial analysis of

multibus multiprocessors. In *Proceedings of the 1984 International Conference on Parallel Processing*, pages 225-227, 1984.

- [3] Laxmi N. Bhuyan, Bao-Chyn Liu, and Irshad Ahmed. Analysis of min based multiprocessors with private cache memories. In *Proceedings of the 1989 International Conference on Parallel Processing*, pages 1-51-1-58, 1989.
- [4] Michel Dubois. Effect of invalidations on the hit ratio of cache-based multiprocessors. In *Proceedings of the 1987 International Conference on Parallel Processing*, pages 255-257, 1987.
- [5] Michel Dubois, F.A. Briggs, I. Patil, and M. Balakrishnan. Trace-driven simulations of parallel and distributed algorithms in multiprocessors. In *Proceedings of the 1986 International Conference on Parallel Processing*, pages 909-916, 1986.
- [6] Michel Dubois and Faye A. Briggs. Effects of cache coherency in multiprocessor. *IEEE Transactions on Computers*, 31(11):1083-1099, 1982.
- [7] Michel Dubois and Jin-Chin Wang. Shared data contention in a cache coherence protocol. In *Proceedings of the 1988 International Conference on Parallel Processing*, pages 146-155, 1988.
- [8] Keki B. Irani and Ibrahim H. Onyuksel. A closed-form solution for the performance analysis of multiple-bus multiprocessor systems. *IEEE Transactions on Computer*, C-33(11):1004-1012, 1984.
- [9] T. N. Mudge and H. B. Al-Sadoun. Memory interference models with variable connection time. *IEEE Transactions on Computers*, 33(11):1033-1038, 1984.
- [10] T. N. Mudge and H. B. Al-Sadoun. A semi-markov model for the performance of multiple-bus systems. *IEEE Transactions on Computers*, 34(10):934-942, 1985.
- [11] Jin-Chin Wang and Michel Dubois. Performance comparison of cache coherence protocols based on the access burst model. *Computer Systems Science and Engineering*, 5(3):147-158, 1990.
- [12] 城和貴. A SURAの解析モデル. Technical Report 93-ARC-99-17, 情報処理学会ARC, 1993.
- [13] 城和貴 and 福田晃. 並列計算機の解析モデル—シミュレーションとの比較—. Technical Report 93-ARC-100-3, 情報処理学会ARC, 1993.
- [14] 城和貴 and 内藤潤. セミ・マルコフ過程を用いたA SURAクラスタのモデル化. Technical Report 92-ARC-97-9, 情報処理学会ARC, 1992.