

ゲーム木の並列探索のための分散共有ハッシュ機構の設計と実現

長島 紀子[†] 中山泰一 野下浩平

電気通信大学 情報工学科

ゲーム木を探索する場合、異なる枝に同一局面が発生することが頻繁に起きる。この探索の重複を避けるため、局面表(ハッシュ表)に局面を登録し、計算結果を再利用するという手法がよく用いられる。ゲーム木の並列探索するときにも、上記のハッシュ表をプロセッサ間で共有できれば、異なるプロセッサ上で計算した結果が利用でき、計算時間の短縮が期待できる。本研究では、まず、分散メモリ型並列計算機(NEC Cenju-3)上に分散共有ハッシュ機構を設計・実現した。続いて、具体的なゲーム木探索問題としてオセロゲームの先手必勝後手必勝の決定問題を取り上げ、分散共有ハッシュ機構の評価実験を行った。実験の結果、たとえば7×5盤のオセロゲームでは、分散共有ハッシュ機構の導入により約30%計算時間が短縮されることが示された。

Design and Implementation of a Distributed Shared Hashing Mechanism for Searching Game-Trees in Parallel

Noriko Nagashima[†], Yasuichi Nakayama and Kohei Noshita

Department of Computer Science,

The University of Electro-Communications

noriko@igo.cs.uec.ac.jp

This paper proposes a distributed shared hashing mechanism for searching game-trees in parallel on distributed-memory computers. If a parallel program for solving game problems uses hash (transposition) tables shared among all processors and if the overhead caused by interprocessor communications is relatively small, then its computation time can be reduced. We have designed and implemented a distributed shared hashing mechanism on the distributed-memory multiprocessor 'NEC Cenju-3', and have solved $M \times N$ Othello (Reversi) problems. The experimental results show that the computation time has been considerably reduced when our hashing mechanism is employed; e.g. for 7×5 Othello problem, reduced by 30%.

1 はじめに

チェス、将棋、囲碁などのゲーム木を探索する場合に並列処理を行なうことが注目されている。先日人間のチャンピオンに勝ち越したチェスプログラム DeepBlue においても、専用の並列処理ハードウェアを用いて並列計算を行なっ

ている。また、筆者らはこれまでに、ネットワークにより結合された UNIX ワークステーション群を利用して、具体的なゲーム木として詰将棋問題を並列に探索し、並列計算による性能向上を示してきた [1][2]。

ゲーム木を探索する場合、異なる枝に同一局面が発生することが、頻繁に起きる。この探索の重複を避けるために、局面表 (transposition

[†] 現在、NEC コンピュータソフトウェア事業本部
Nagashima's current affiliation: NEC Corporation

table) に局面を登録し、のちに同一局面が発生したときに、局面表の結果を再利用するという手法がよく用いられる。局面表はハッシュ表を用いて実現する。

本研究では、ゲーム木の並列探索に、上記のハッシュ表をプロセッサ間で共有することに着目した。ハッシュ表の共有により、異なるプロセッサ上で計算した結果を利用でき、計算時間の短縮を期待できる。

そこで、まず、分散メモリ型並列計算機上に、ハッシュ表をプロセッサ間で共有するための機構(分散共有ハッシュ機構)を設計・実現した。分散メモリ型並列計算機において、プロセッサ間でデータを共有し、あらゆる並列プログラムを正しく動作させるためには、厳密なデータの一貫性の制御が求められ、大きなオーバーヘッドがかかる。これに対し、ゲーム木の探索におけるハッシュ表に登録するデータの特徴より、ゲーム木の並列探索は、緩いデータの一貫性で済ませることができ、小さなオーバーヘッドで分散共有ハッシュ表を導入できる。

続いて、具体的なゲーム木探索問題として、オセロゲームの先手必勝後手必勝の決定問題を取り上げ、この分散共有ハッシュ機構を評価する実験を行った。この実験の結果、分散共有ハッシュ機構を導入することにより、導入しない場合と比較して、大きな性能向上を得た。たとえば、 7×5 盤のオセロゲームの先手必勝後手必勝の決定問題においては、導入しない場合には18時間要する実行時間が、導入した場合には13時間となり、約30% 実行時間の短縮となった。

2 分散共有ハッシュ機構の提案

ゲーム木において、別の枝に同一局面が発生することは、頻繁に起こる。そこで、ゲーム木の探索では、局面をハッシュ表(局面表)に登録しておき、同一局面に対する再計算を防ぐ方法が、一般的に用いられている。これにより大幅に計算時間の短縮を行なうことができる。また、同一局面に限らず類似局面についてもハッシュ表に格納されているか検索を行ない、その情報を利用することにより、計算時間の短縮を図る

ことが可能であると報告されている [3][4]。

ここで、互いに独立したハッシュ表を備えたプロセッサ (P_x, P_y, P_z, \dots) が、図1のような探索木を並列に探索する場合を考える(ただし、D は同一の部分木)。

ハッシュ表の検索により、各プロセッサ単位では、探索の重複はない。しかし、全体を一つの探索木とすると、プロセッサ間で探索の重複が行なわれている。

この重複を避けるためには、プロセッサ間でハッシュ表を共有すれば良い。これにより、計算時間の短縮につながる可能性がある。

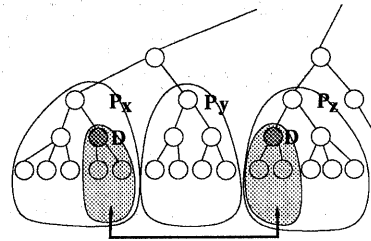


図1: プロセッサ間の同一・類似局面

2.1 ハッシュ機構の性能向上の見積り

本分散共有ハッシュ機構は、プロセッサ間通信を用いることにより、プロセッサ間でハッシュ表を共有し、大規模なハッシュ空間を実現している。すなわち、検索、登録要求はプロセッサ間通信を必要とする。よって、「ある局面の結果を得るための時間(試行時間)」は、「プロセッサ間の通信時間(C)」「局面の結果を得るためのゲーム木の探索時間(M)」「ヒット率(α)」の3つの要素から推定することが可能である。実際には C は、通信時間のほか検索時間を含む待ち時間である。分散共有ハッシュ機構を使用した場合の一回の平均試行時間は次のようになる。

$$C + (1 - \alpha) \times M$$

分散共有ハッシュ機構を使用しない場合の平均試行時間は M である。したがって、以下の関係を満たす C, M, α の下であれば、本分散共

有ハッシュ機構の導入は、性能向上を得るといえる。

$$C + (1 - \alpha) \times M < M$$

$$\Leftrightarrow \frac{C}{\alpha} < M$$

たとえば、 C 、 M 、 α が $C = 5msec$ 、 $M = 100msec$ 、 $\alpha = 0.1$ のとき、ゲーム木の並列探索は分散共有ハッシュ機構を導入することにより性能向上を得ると見積もることができる。

2.2 データの一貫性

汎用の分散メモリ型並列計算機において、多数のプロセッサがメモリを共有するためには、一般的に、厳密な一貫性 (sequential consistency[5]) の保持が必要であり、オーバーヘッドは極めて大きなものとなる。

そこで、本研究では、次にあげるゲーム木探索のためのハッシュ表へ登録するデータの特徴に注目した。

特徴 1 ある局面に対する情報は、どのプロセッサが計算しても同一の結果を得る

特徴 2 ハッシュ表に、必要な情報がない場合には、改めて計算すれば、どのプロセッサでも正確な結果を得ることができる

この特徴により、ゲーム木の探索に用いられるハッシュ機構は緩いデータの一貫性で済み、オーバーヘッドを小さくすることができる。

3 分散共有ハッシュ機構の構成

分散メモリ型並列計算機上で分散共有ハッシュ機構をシステム・ソフトウェアとして設計、構成する。システムは、ゲーム木の探索をする「ユーザ・プログラム」とプロセッサ間で共有するハッシュ表を提供する「システム・プログラム」から成る。システムの概略を図 2 に示す。

ユーザ・プログラムが、同時に動作している全プロセッサのハッシュを参照することにより、大規模なハッシュ表が実現される。ユーザ・プログラムとシステム・プログラムはコンテクス

ト切替によるオーバーヘッドを防ぐため、メモリ領域を共有する軽量プロセスとして実現し、並行に動作している。

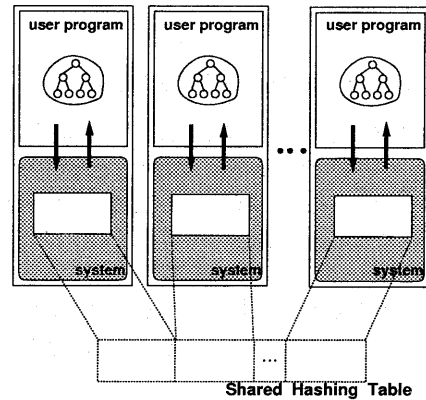


図 2: システムの概略

ユーザ・プログラムの要求はシステム・プログラムによって処理される。要求を依頼したプロセッサを「依頼プロセッサ」、実際の処理を担当するプロセッサを「担当プロセッサ」とする。

● 探索要求

- 1 ハッシュ・キー (key) から担当するプロセッサを決定する。
- 2 担当プロセッサへデータを送信して、「検索」を依頼する。
- 3 担当プロセッサからの検索応答を受け、ユーザ・プログラムに結果応答する。

● 登録要求

- 1 key から担当するプロセッサを決定する。
- 2 担当プロセッサへデータを送信して、「登録」を依頼する。

ただし、依頼プロセッサと担当プロセッサが同一の場合は、依頼プロセッサのハッシュ表の検索および、ハッシュ表への登録を行なうものとする。

この基本的な処理方法により、ハッシュ表の検索時に最新の情報を得ることができる。

4 性能向上のための方法

前章において述べた処理方法を基本にして、緩い一貫性を保つ、検索および登録要求の処理方法を示す。

4.1 データの複製を保持する方式

他のプロセッサへ要求を依頼する場合には、データの送受信によるオーバーヘッドがかかる。

そこで、このオーバーヘッドを削減するために、依頼プロセッサの持つハッシュ表に、データの複製を保持する方式を提案する。

データの複製を保持する方式により、検索時に依頼プロセッサの持つハッシュ表に複製データが存在するならば、ユーザ・プログラムに速いレスポンスで検索結果を返すことができる。

また、2.2 節において述べた、ゲーム木探索の特徴より、複製データの無効化の処理、複製データをどのプロセッサ所有しているかの管理をする必要なく、複製データの一貫性を保持できている。

4.2 タイムアウトを利用する方式

前節の処理方法を採用した場合においても、依頼プロセッサと担当プロセッサが異なる場合には、検索応答時にはプロセッサ間通信によるオーバーヘッドがかかる。

これを防ぐために、「タイムアウト時間」の設定ができるようにする。担当プロセッサへ検索を依頼してから、設定された「タイムアウト時間」を過ぎても担当プロセッサからの検索応答がない場合、担当プロセッサからの応答を待たずに、依頼プロセッサは情報がなかったと判断し、ユーザ・プログラムに「情報が存在しない」と結果応答する。

以上の処理方法を採用する場合においても、情報が存在しないと応答したとしても、前述の通り各プロセッサが改めてその局面を計算すれば良いので、ユーザ・プログラムの動作へ支障を来すことはない。これによりシステム・プログラムの検索応答のかかり過ぎによるユーザ・プログラムのサスペンドを防ぐことができる。

4.3 プライオリティを利用する方式

本分散共有ハッシュ機構を用いてハッシングを行なう場合、大規模なハッシュ表を実現しているため、大幅なゲーム木の探索回数の減少になる。2.1 節において述べたように、以下の関係式を満たす C , M , α の下であれば、本分散共有ハッシュ機構を導入することによる性能向上を得ることが可能であると見積もることができる。

$$\frac{C}{\alpha} < M$$

しかし、 M が小さすぎると、性能向上を得ることができない。

そこでプライオリティを利用して、 M が大きなものを共有するようにする。

本研究における、プライオリティのつけ方は、以下の通りである。 M の大きい局面 (ハッシュ機構の利用価値の大きい局面) には、高いプライオリティをつける。また、 M の小さい局面 (ハッシュ機構の利用価値の低い局面) には、低いプライオリティをつける。

また、分散共有ハッシュ機構に限らず、一般にハッシュ機構において、ハッシュ表が溢れた時、登録したデータ全てを消去してしまうと、データが全て無駄になってしまう。そこで、本分散共有ハッシュ機構では、ハッシュ表が溢れた時には、表全体を空にするのではなく、このプライオリティを用いて、できるだけ利用価値の高いデータを残すようにする。

5 ゲーム木による性能向上実験

2章から4章で述べてきた分散共有ハッシュ機構を設計・実現し、分散メモリ型並列計算機上で、評価実験を行なった。

5.1 計算環境

分散メモリ型並列計算機 Cenju-3(NEC) を使用した。Cenju-3 は、マイクロプロセッサ V_R4400 を要素プロセッサとし、疎結合型で実装されている。最大接続可能 PE 数は 256 である。ネットワークには多段接続網を使用して、

40MB/sec を実現している [6]。ハードウェアの概略図を図 3 に示す。

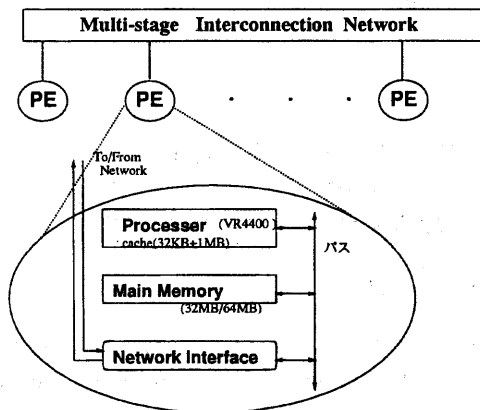


図 3: Cenju-3 のハードウェア構成図

5.2 実験対象

ゲーム木の問題として「 $M \times N (M \geq N)$ 盤のオセロゲームの先手必勝後手必勝の決定」を取り上げる。ゲームの探索木の初期盤面から数手を展開して、各プロセッサに一つずつ部分木を割り当てる。表 1 に示す各 A~F 構成についての総実行時間の比較を行なう。

表 1: 構成

構成	ハッシュ表	要求の処理方式
A	非共有	基本方式
B	共有	厳密なデータの一貫性制御
C	共有	基本方式
D	共有	複製を持つ
E	共有	プライオリティ
F	共有	D + E

5.3 実験結果

各 A~G 構成についての総実行時間を 6×5 盤、 7×5 盤につて、表 2、表 3 に示す。

- 6×5 盤: PE 数 16 として実行した。A 構成では 48 分要したのに対して、F 構成では 39 分となり、9 分の総実行時間の短縮となり、約 20% の性能向上となった。

表 2: 6×5 盤の総実行時間

構成	総実行時間 (sec)
A	2872
B	3003
C	2903
D	2857
E	2433
F	2366

- 7×5 盤: PE 数 20 として実行した。A 構成は 18 時間要したのに対し、F 構成は 13 時間となり、5 時間の総実行時間の短縮となり、約 30% の性能向上になった。

表 3: 7×5 盤の総実行時間

構成	総実行時間 (sec)
A	64203
B	63379
C	61703
D	61189
E	65638
F	47640

ハッシュ表を共有しないハッシュ機構 (A 構成) と比較して、分散共有ハッシュ機構の導入は、優れた性能向上を示した。これより、本分散共有ハッシュ機構は、プロセッサ間の通信によるオーバーヘッドの削減を実現し、大規模なハッシュ表を所有する効果を発揮することができるといえる。

さらに、「 6×5 盤」を対象とした実験と比べ、「 7×5 盤」を対象にした実験は、「A 構成と比較して F 構成の性能向上率」が良いことがわか

表 4: $M \times N$ 盤オセロゲームの勝敗表

(● : 先手勝ち, ○ : 後手勝ち, † : J. Feinstein)

	4	5	6	7	8	9	10	11	12
4	○	●	●	●	●	●	●	●	?
5		●	●	●	●	?			
6			○†						

る。これより、探索木の規模が大きいほど、本分散共有ハッシュ機構導入により、総実行時間の短縮につながるという。

また、具体的な問題として、オセロゲームに限らず、将棋、チェスなどの様々なゲーム木の並列探索において、それぞれの探索に適する要求の処理方法を採用することにより、本分散共有ハッシュ機構導入が、総実行時間の短縮につながることを期待される。

6 まとめ

本研究は、分散メモリ型並列計算機上で、分散共有ハッシュ機構を導入することによる性能向上を期待できる条件を見積り、また、ゲーム木探索のデータの特徴から緩いデータの一貫性を保持することで分散共有ハッシュ機構の性能が保証されることを示し、性能向上のためのいくつかの方法を設計、実現をした。

さらに、実現した分散共有ハッシュ機構を用いて、実際のゲーム木の問題として「 $M \times N (M \geq N)$ 盤のオセロゲームの先手必勝後手必勝の決定」を用いて実験を行った。その結果、非共有のハッシュ機構と比較して、本分散共有ハッシュ機構は大幅な総実行時間の短縮を示した。これにより、本分散共有ハッシュ機構は、プロセッサ間の通信のオーバーヘッドを削減し、大規模なハッシュ表を保持していることを有効的に活用しているといえる。また、実験の結果より、大規模な探索木になるほど、本分散共有ハッシュ機構の性能が発揮できることが示された。

なお、筆者らが求めた $M \times N$ 盤のオセロゲームに関する結果は表 4 の通りである [7]。ただし、 6×6 盤は Feinstein の報告による [8]。

謝辞 分散メモリ型並列計算機 Cenju-3 の利用環境を御提供頂いた、NEC C&C メディア研究所 並列処理センターに感謝致します。

参考文献

- [1] Y.Nakayama, T.Akazawa and K.Noshita, "A Parallel Algorithm for Solving Hard Tsume-Shogi Problems," ICCA Journal, Vol.19, No.2, pp.94-99, June 1996.
- [2] 中山 泰一, 赤澤 忠文, 野下 浩平, "ゲーム木の並列探索のための分散的実行管理機構," 電子情報通信学会論文誌, Vol.J79-D-I, No.9, September 1996.
- [3] 松原 仁 編著, "コンピュータ将棋の進歩," 共立出版, 1996.
- [4] D.N.L.Levy(Ed.), "Computer Games I," Springer-Verlag, 1988.
- [5] L.Lamport, "How to Make a Multiprocessor Computer that Correctly Executes Multiprocess Programs," IEEE Trans. Comput., Vol.C-28, No.9, pp.690-691, September 1979.
- [6] 広瀬 哲也, 細見 岳生, 丸山 勉, 加納 健, "並列コンピュータ Cenju-3 のプロセッサ間通信方式とその評価," 情報処理学会論文誌, Vol.37, No.7, pp.1378-1387, July 1996.
- [7] 長島 紀子, "ゲーム木の探索のための分散共有ハッシュ機構," 電気通信大学大学院電気通信研究科情報工学専攻修士論文, February 1997.
- [8] J.Feinstein, "Amenor Wins World 6×6 Championships!," British Othello Federation's Newsletters, July 1993.