

ゲーム木の並列探索のための分散的実行管理機構の設計と実現

4U-7

赤澤 忠文, 中山 泰一, 野下 浩平 (電気通信大学情報工学科)

1 はじめに

現在は多数の計算機がネットワークで接続されており、遠隔マシン上でプロセスを実行させることも可能である。筆者らはこのような分散的環境を利用して単一の計算機では時間のかかるようなゲーム木の探索を並列に実行することを目的とする。

これを達成するためには、小問題を解くためのプロセスを遠隔マシン上で自動起動する機構(並列実行機構)と遠隔マシン間でハッシュ表を共有する機構(共有ハッシュ機構)が必要とされる。本稿ではこの2つの機構の設計・実現と、ゲーム木探索の並列化による速度向上に関する計算実験について述べる。

2 設計

2.1 計算機環境

現在対象としているのは、数十台程度のネットワーク接続されたUNIXワークステーション群である(複数の機種が入り混じっていても構わない)。筆者らが行なった実験にはノードサーバとして7台のNEWS5000がFDDIで接続され、それぞれのノードサーバから十数台のNEWS3000がEthernet接続された環境を使用した。

2.2 並列実行機構

並列にゲーム木を探索する場合、木を展開し小問題に分割する部分(マスタープロセス)と与えられた小問題を解く部分(スレーブプロセス)に分ける(図1)。マスタープロセスに対しては小問題を解かせるべきスレーブプロセスを起動し実行

要求を与えられること、スレーブプロセスに対しては計算し結果をすぐ返すことが要求される。またスレーブプロセスには実行中にマスタープロセスからの要求により実行を中断・初期化するような機能(リセット)も必要である。

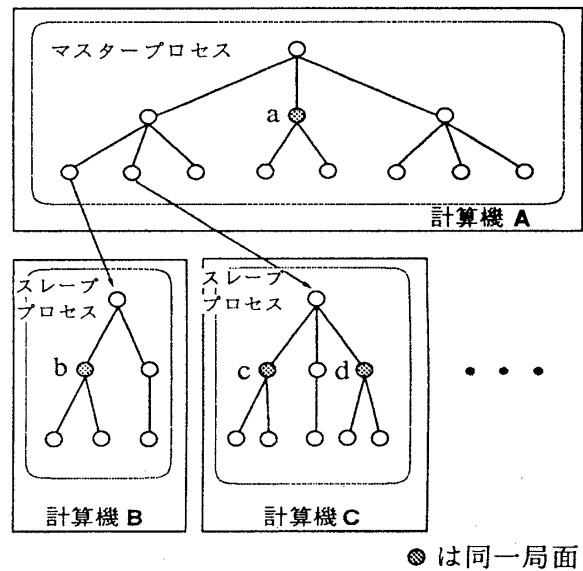


図1: ゲーム木の構成

小問題は数が多く、マスタープロセスは遠隔マシンそれぞれに多数の実行要求を与えることになる。そこでアクティビティ方式並列実行機構 [1] を応用して、あらかじめスレーブプロセスを1つ作成しそれを繰り返し再利用する設計とした。スレーブプロセスの自動起動には rexecd を利用し、スレーブプロセス側では標準入出力を読み書きすることでマスタープロセスと通信することができる。また、スレーブプロセスが標準エラー出力に送ったメッセージはマスタープロセスの標準エラー出力に送られるようにした。更に本並列実行機構ではマスタープロセスからスレーブプロセスに割り込みをかけて実行を中断・初期化する機構を作成した。設計の概要を図2に示す。

Design and Implementation of a Distributed Execution Mechanism for Searching Game-Trees in Parallel by TADAFUMI AKAZAWA, YASUICHI NAKAYAMA and KOHEI NOSHITA (Department of Computer Science, The University of Electro-Communications).

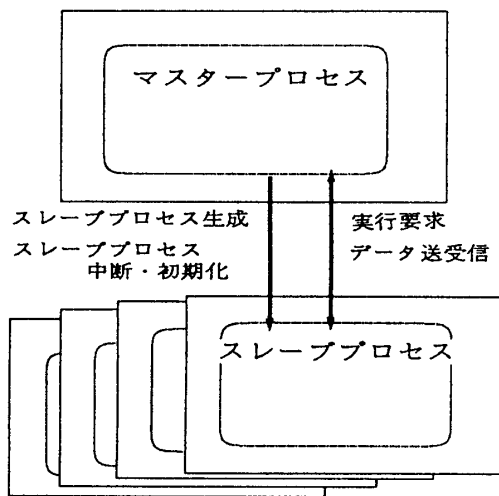


図 2: マスタープロセスとスレーブプロセス

2.3 共有ハッシュ機構

ゲーム木を探索するとき同一(または類似)局面が現れることがあり、その都度再計算するのは非常に無駄である。チェスの局面表に見られるように再計算を防ぐ方法としてハッシュ表を用いるのはゲーム木探索の基本的技法となっている [2]。

2.2の並列実行機構によりゲーム木を探索する場合マスタープロセス及びスレーブプロセスが個々にハッシュ表を持つことはできる。しかしスレーブプロセス間の共有ハッシュ機構を提供することにより更なる計算効率の向上が見込まれる。

例えば図 1 において a~d が同一局面であるとき、共有ハッシュ機構がない場合には各計算機において a と b と c を計算してしまうことがある。共有ハッシュ機構を用意すればこれらは最初の 1 回だけ計算するだけでよくなる。

本共有ハッシュ機構の一貫性制御は [3] の方法を用いた。また、ゲーム木では一般的なワークステーションの主記憶上に割り当てられないほど大きな領域がハッシュ表のために必要となる。本機構ではディスク上のファイルを使うことによりこれを実現している。

3 計算実験

ゲーム木の例として詰将棋を取り上げる。逐次的プログラムによる木の探索法としては縦型探索と横型探索がある。縦型探索は解答率が高いが時

間がかかり、横型探索は速いが解答率が低い。詰手数が 19 手以上の問題になると、1 台の計算機で縦型探索を用いた場合短時間では解けないものが多い。

詰手数が 19~25 手の 50 問題について実験を行った(詰パラ 92 年度短大)。普通のワークステーション上で最も高性能なもの 1 つである逐次的プログラム T2 によると 49 題解けるが 5 割以上は 3 時間以上かかる [2]。最大 64 台の計算機で 64 個のスレーブプロセスを起動させた結果は次の通りである。

- 50 題中 41 題が約 1 時間以内で解けた。
- 50 題中 32 題が約 10 分以内で解けた。

上記のうち 64 台での性能向上が 100 倍以上であった 5 問題を表 1 に示す。詳しい実験結果については別途報告する(第 1 回ゲームプログラミングワークショップ 1994)。

表 1: 実験結果(単位:秒, 括弧内は性能向上比)

問題	1 台 T2	16 台	64 台
前期 2	55000(1)	72(763)	66(833)
23	243000(1)	6496 (37)	2413(100)
後期 3	16200(1)	229 (70)	155(104)
11	10200(1)	59(172)	33(309)
24	23100(1)	90(256)	167(138)

今回の実験ではマスタープロセス及びスレーブプロセス内でのハッシングは行なっている。しかし共有ハッシュ機構は使用していない。今後、共有ハッシュ機構を使用することにより性能の向上が期待される。

参考文献

- [1] 中山, 永松, 出口, 森下: 共有メモリ型並列機のための新しいアクティビティ方式並列実行機構, 情報処理学会論文誌, Vol.34, No.5, pp.985-993 (1993).
- [2] 伊藤, 野下: 詰将棋を速く解く 2 つのプログラムとその評価, 情報処理学会論文誌, Vol.35, No.8, pp.1531-1539 (1994).
- [3] 中山, 赤澤: ユーザ・レベル・ライブラリのみによる分散共有メモリ・サーバ構築の試み, 第 47 回情報処理学会全国大会論文集, 5B-5 (1993).