

将棋における局面探索の多重並列計算

柳 圭二郎^{†1} 柴原 一友^{†2} 但馬 康宏^{†2} 小谷 善行^{†2}

概要

コンピュータ将棋の着手を決定するために最も時間がかかるのは局面探索であり,探索の速度を向上させれば強化につながるとされている. また, 近年コンピュータ将棋で注目されているのが複数のコアを持つ CPU による並列計算である. そこで, この局面の並列探索を, 並列処理に特化した CPU である Cell Broadband Engine (以下 Cell)を用いて行うとどれだけの効果があるかを実験する. 並列化アルゴリズムは既存のものを用いるが, Cell が並列探索に有効かどうかを調べる. 最終的には Cell 上に将棋プログラムを実装し, 既存のプログラムとの対局の勝率を求める.

Multiple Parallel Search in Shogi

Keijirou YANAGI Kazutomo SHIBAHARA Yasuhiro TAJIMA Yoshiyuki KOTANI

Tokyo University of Agriculture and Technology

Abstract

It takes the time to decide a best move by position search in Shogi, so improving the speed of the search leads to strengthening. Recently, it pays attention to the parallel processing by CPU with many cores. Then, there is a Cell Broadband Engine (Cell) that specializes in the parallel processing. It experiments on this parallel search by using Cell. The search that uses Cell has not had the example. The algorithm of making to the parallel uses the existing one. But, it is examined whether Cell is effective to a parallel search. Finally, it mounts on the Japanese chess program, and experiments the chance of success of the play.

^{†1} 東京農工大学 工学府 情報工学専攻 小谷研究室

^{†2} 東京農工大学 共生科学技術研究院 システム情報科学部門

1. はじめに

近年のコンピュータの発展によるマルチコア CPU のマシンが多く開発され、その性能を最大限に用いるために複数のコアで同時に計算を行う並列処理が研究されている。コンピュータ将棋においても探索の速度上昇の目的で、並列探索は注目されている。そこで、並列処理に特化した CPU である Cell (セル) [1]が局面の並列探索に有効であるかどうかを実験する。Cell は1つの PPU と8つ (ユーザが任意に使えるのは6つ) を持つヘテロジニアスなマルチコア構成をしている。(図1)この複数のSPU が並列処理に強く、独自の構造を持つ。

この中心となる PPU と、作業のための SPU を分担して並列化に用いることにより、同じように分割するより効率的な仕事の配分法を提案する。

この実験では Cell を扱う環境として、PlayStation3 上に Fedora core 6 (Linux) を載せ、その上に将棋プログラムを置く形を用いる。

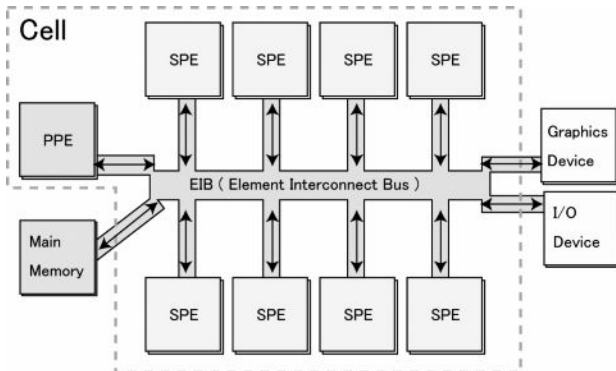


図1 Cell の構造

2. 局面探索の並列処理の関連研究

以前より並列処理はコンピュータチェスの分野で多く研究されており、初期の良い結果を残したアルゴリズムが改良されている。これらが Cell で効果的に稼働するかどうかを実装して比較対象とする。

・PV-Split [1]

近年のゲーム木の並列探索において基本となっているアルゴリズム。探索の最初に

Principal Valiation ノードを探索することで効率的な枝刈りを行える。

・YBWC(Young Brother Waiting Concept) [2]

PV-Splitting を改良したアルゴリズム。「すべてのノードにおいて最初の子ノードは、他の子ノードを探索する前に評価しなければならない」という考え方に基づいた手法。

本研究では上記の PV-Split と YBWC を元にして、負荷分散の新たな手法を試す。

3. 並列探索における問題

ゲーム木の局面探索を並列に行うときに以下のような問題がある。

- ・探索順序
- ・負荷分散
- ・情報共有

まず探索順序による問題は、 $\alpha\beta$ アルゴリズムは探索窓によって枝刈りを行うが、その窓の値によっては効率的な枝刈りができない。並列に探索する個々のノードが窓の値によって枝刈りの効率が変わるので探索の順序による窓の更新が重要である。

また、負荷分散は、分割したノードの部分木がすべて同じ大きさであるとは限らないので、均等に分配しても待機状態になるコアが出てしまう。そこで待機となったコアは他の処理中のコアから待機中の仕事を取得することや(ワークスチーリング), ある一定の規則で仕事を受け渡す仕組みが必要である。

情報共有の主なものはトランスポジションテーブルである。共有するデータに関して、並列処理中に同時に書き込みを行うと、重大なエラーが起こる可能性がある。この対処法には書き込む対象をロックして他に書き込ませないようにするなどがある。

4. 並列の多重計算

並列計算を多重に行い効果を発揮する環境とそのアルゴリズムについて説明する。

4. 1 多重計算が効果を発揮する環境

局面探索の並列処理において、負荷分散が必要になる場合は、処理をするコア数が可能手数を上回ったときである。このとき、何もしていない待機中コアにいかにか処理させるかが問題となる。

また、近年のコンピュータ将棋において多く用いられている実現確率探索[4][5]がある。この実現確率探索は、学習データの棋譜から、ある着手のカテゴリが指される確率を求め、その確率が一手ごとに積算され、ある閾値を超えた着手の探索を打ち切るものである。

通常の探索では、この打ち切りが多く生じ、経験的に良いとされる着手が深く探索される。ここで探索する着手が少なくなり、待機中のコ

ア数が多くなる。一般的に並列探索は、横に広く同時に行うため、全幅探索に向いているとされている。しかし、棋譜データに頼るものの、効率が良いとされている実現確率探索は、並列処理の多重化に向いていることが考えられる。

これらの負荷分散に注目し、新しい仕事の分配方法を提案する。

4. 2 多重並列による不可分散

局面探索の並列処理の基本は YBWC を基にする。これは同期をとることで安全性があり、使いやすく、多くの派生が研究されているためである。

この中で独自の手法として、並列化のために分割したノードにおいて、さらに負荷分散のために多重に並列化する。

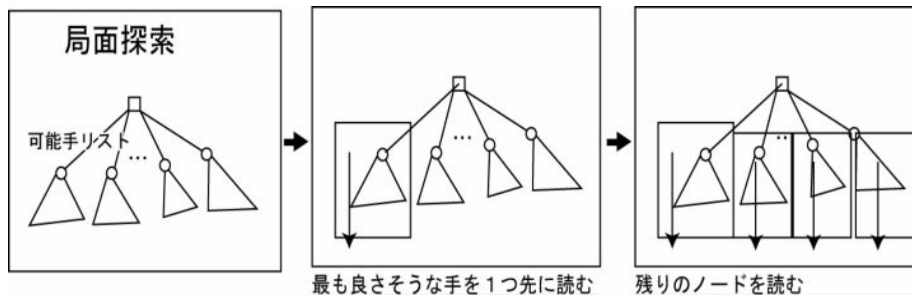


図2 YBWCの流れ

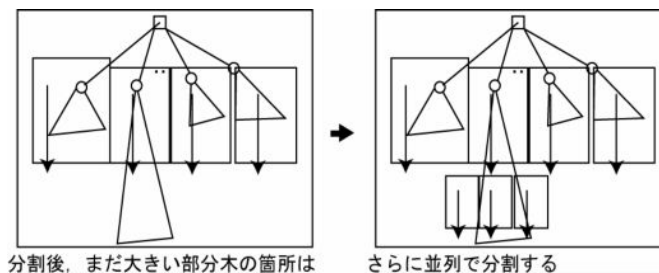


図3 局面の多重分割

基本的な分割は、各探索ノードの可能手ごとに行う。そして可能手の先の評価値を求め、 α β 探索の枝刈りを行う窓の値を更新する。探索を並列化することにおいて、ノードごとに分割された処理を一つの仕事として各コアが処理する。

将棋における平均的な可能手数に比べ、コア数が多く存在するので、一時的に何もしない待機中コアが出てしまう。ここで探索を実行しているコアが自分の処理をノードごとに分割し、

待機中のコアに分配する。処理中に次の処理をキュー状態で保持しておく。この配分によってはコアごとに偏りが出てしまうが、並列に探索を行っているときに一定の深さを超えて探索が続いているノードがあるとき、その探索を分割してコアのキューに入れる手法をとる。

4. 3 小グループ型の負荷分散

また、並列に分割するに際し、小さいグループに分け、さらにその中で並列化することがで

きる。(図4)

Cellは、メインの処理をPPUが行い、並列に分割する仕事をSPUが行う手法を用いることが良いとされている。そこで、CellをPPUとSPUの小グループとしてひとまとまりとし、それをさらにクラスタの形で並列に連結して

並列処理を行うことを考える。このとき、小グループの中のPPUが、他のグループとの仕事の受け渡しをすることや、SPUへの細かな配分を設定する。中間の通信処理を任せることで各SPUは細分化された処理を行うだけでよく、効率化が期待できる。

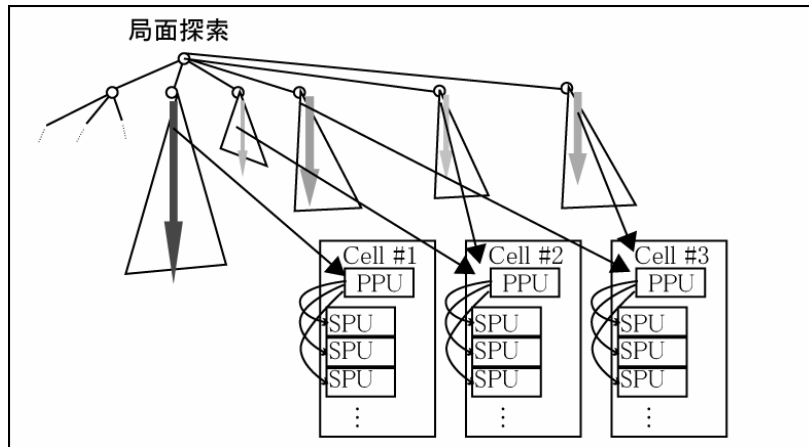


図4 小グループでの割り振り

5. 実験

この実験では Cell を扱う環境として、PlayStation3 上に Fedora core 6 (Linux) を載せ、その上に将棋プログラムを置く形を用いる。本研究室で開発された将棋プログラム「まったりゆうちゃん」を Cell に移植し、並列化を行う。この並列化に際し、いくつかの種類を実装し、比較をする。

実験に用いるのは以下の3種類を用いる。

- 並列無し(逐次 比較用)
- 並列有り(Cell上での PV-Split 比較用)
- 並列有り(小グループ型)

並列無しは PPU 1 つのみで動くもので比較に用いる。並列有り(PV-Split)も、並列探索の基本である PV-Split のみを用いる。並列有り(小グループ型)が本実験で結果を出したいものである。

最終的には、既存の将棋プログラムと対局して、その勝率を評価とする。

6. おわりに

現在のマルチコアマシンでは、局面探索の分

割数が多いとき、各コアの待ち時間が生じない。この環境では、まだ本手法が十分な効果を出すことができない。しかし今回の実験から、ノードごとに分割することができれば、マシン台数自体を並列につなぐクラスタ化に応用し、効果を発揮できるようになる。

ゲーム木の並列局面探索に一般的な手法は、ハードウェアのアーキテクチャによっても効果が異なると思われる。それを踏まえても、並列探索が効率的であれば、ハードウェアの発展がゲームの強さに大きく貢献できるはずである。

参考文献

- [1] T. A. Marsland and M. S. Campbell. Parallel Search of Strongly Ordered Game Trees. ACM Computing Surveys, 14(4):pp533-551, 1982.
- [2] R. Feldmann, P. Mysliwicz, and B. Monien. Game tree search on a massively parallel system. Advances in Computer Chess 7, pp203-219, 1993
- [3] Mark G. Brockington. Asynchronous Parallel Game-Tree Search. PhD thesis, University of Alberta, Edmonton, Canada, 1998
- [4] CELL, Cell Broadband Engine, Sony Computer Entertainment, <http://cell.scei.co.jp>