GTC Workshop Japan 2011 (2011 年 7 月 22 日)

min-max ゲーム木およびモンテカルロ木探索の GPU 上への実装

An Inprementation of Search Algorithms on GPU
for min-max game tree and Monte Carlo Game Tree

渥美 清隆
Kiyotaka ATSUMI

鈴鹿工業高等専門学校 情報処理センター (〒 510-1326 三重県鈴鹿市白子町)

Key Words: *Min-max Game Tree, Monte Carlo Game Tree, OpenCL, SIMD Operation*

1 はじめに

二人零和有限確定完全情報ゲームに関する研究は古くから行われており，ゲーム木を効率的に探索する $\alpha\beta$ 狩りや MTD 等の様々なアルゴリズムも開発されてきた．それでもより深く探索するためには多くの計算資源が必要であり，多くの計算資源つまり並列計算を利用するためのアルゴリズムの開発も行われている．

ほとんどは複数のプロセッサを内蔵した大量の計算機を Infiniband などの専用通信システムで接続したクラスタ型並列計算機の利用を前提として，ゲーム木の枝を分割してプロセッサに割り当てる方法が提案されている．この場合，プロセッサ間で展開した節の情報を交換して同じ局面の冗長な展開をできる限り抑えることに腐心している．特に分散メモリ環境ではプロセッサ間通信が，プロセッサの演算能力に比べて極端に遅くなってしまうため，多少の効率性の低下と引換に並列性を向上させる試みがなされている．

一方，GPU や Cell を用いた並列計算の研究も進められている．クラスタ型並列計算機と比べると，プロセッサあたりの性能比では劣るものの，GPU では 1LSI に 100 以上のプロセッサ載せられるメリットを生かしたアルゴリズムが検討されている [1, 2].

本研究ではオセロゲームを対象として，$\alpha\beta$ 狩りを用いた min-max ゲーム木とコンピュータ囲碁プレイヤにおける実装の実績があるモンテカルロ枝刈りを用いたゲーム木およびその両方の特徴を持つアルゴリズムを単一ノード上の AMD Radeon HD5970x2(GPU としては 4 基，6400 プロセッサ) に OpenCL で実装し，探索効率や勝率などについて検討する．

2 実装方法

最初のゲーム木の展開は CPU で行い，一定程度の節数に分割された所で，GPU 上のメモリに転送し，GPU で計算を始めさせる．単純な min-max/$\alpha\beta$ 狩りでは，終局または，一定の深さまで探索し，その状態を評価関数で評価して，結果を GPU 上の開始接点に伝える．MonteCarlo 法の場合は，あまりにも不利になると簡単に予測できる手を除く全ての手からランダムに選択して，お互いにプレイアウトするまで打ち合う．これを一定回数実施し，その勝率で開始接点から選択可能な手の評価を行う．両方を合わせた方法では，状態評価に置いて，そこから MonteCarlo 法を用いてプレイアウトまで一定回数ゲームを実施し，勝率を評価値とする．

GPU 上のメモリは決して潤沢とは言えないので，できる限りゲーム状態を記憶する構造体は小さい方が良い．そのため，文献 [1, 2] に慣って，オセロゲームボードは 128bit で表現する．具体的には白石が置いてある場所を表現するために 64bit，黒石が置いてある場所を表現するために 64bit 使い，両方を OR 演算することで，石が置かれていない場所を得ることが出来る．また，128bit は GPU のメモリバス幅の基本単位なので，効率的なメモリ転送が期待出来る．

石が置けるかどうかの判定や，石を反転する作業は，文献 [2] と同様に SIMD 演算を使い，4 方向同時に実行する．当然, この部分での条件分岐は避けて論理演算だけで行い，判定の場合は，結果を論理値で残すようにする．石を反転する作業では，4 方向において，反転させなければならない全ての石について，同時に反転できるようにする．

AMD Radeon HD5800 シリーズでは，SIMD エンジン当りでは 80 器のプロセッサと 32kbyte の Local Data Share を 1 式として備えている．しかし，80 器のプロセッサの中身は FAT プロセッサ 1 器と Thin プロセッサ 4 器の 5 器 1 組として 16 組であり，FAT プロセッサと Thin プロセッサは同時に使えない．そのため，実質的には SIMD エンジン 1 式で 64 スレッドまで動かすことが出来ると考えて良い．但し，Local Data Share が 32kbyte しかないことを考えると，スレッド数の多さに比べて共有メモリが小さいと言わざるを得ない．

そこで，本研究では，SIMD エンジン 1 式では 16 スレッドに抑え，Thin プロセッサ 4 器を SIMD 演算に使うようにすることで，演算資源と共有メモリのバランスを向上させ，全体のスループットを向上させるアルゴリズムを提案する．このアルゴリズムでは，スレッド間で枝刈りの幅や展開済み節を共有でき，節の展開数を抑えることにより，性能向上が期待出来る．

参考文献

[1] K. Rocki and R. Suda: “Parallel Minimax Tree Searching on GPU,” Parallel Processing and Applied Mathematics, Part 1, LNCS 6067 (2009).

[2] 久保田，佐藤，高橋: “マルチコアプロセッサと SIMD 演算によるモンテカルロ木探索を用いたオセロの実装”, 情報処理学会，ゲーム情報学研究会, Vol. 2009-GI-22, No. 7 (2009).