

⑧ A Highly Parallel Chess Program

E.W.Felten and S.W.Otto(California Inst.of Tech.,米国)

発表要旨

我々は、MIMD型のコンピュータ上で実行するチェスのプログラムを開発した。各々の木探索に要する時間は、非常に差があるのでホットスポットにプロセッサパワーを集中させるために動的にプロセッサの負荷割付を行わなくてはならない。木探索にはパラレル $\alpha-\beta$ 刈り込み法を使用し、負荷分散には、再帰的principal-variation-splittingアルゴリズムを使用している。またグローバルハッシュテーブルを実現するために、分散メモリ型のマシンを共有メモリモードで使っている。このプログラムの実行スピードは、512プロセッサのNCUBE hypercubeにおいて控えめにみて単体のときと較べて約170倍になった。このプログラムより10倍遅いバージョンでも、コンピュータチェスの大会で2勝2敗であった。

質疑応答

質問：なぜチームの数が10が最適なのか？

回答：チェスの可能着手数は約40である。チームの数は多すぎても少なすぎてもいけない。10が最適のようである。

質問：グローバルハッシュテーブルを実現するために共有メモリをシュミレートするのだったら、NCUBEでなく共有メモリのマシンでプログラムするのはどうか？

回答：グローバルハッシュテーブルによるオーバーヘッドは2.3%であり、それは、共有メモリのマシンでプログラムをした方がよいという強い理由にはならない。

質問：共有メモリのマシンでもこのチーム構造になりますか？

回答：このチーム構造でよいと思う。

質問：パラレル $\alpha-\beta$ 枝刈りにおいてどのように探索の優先順位をつけているのか？

回答：木探索のノードには、3種類あってそれぞれ個別に探索したり、並列に探索したりする。