




Ponanza アピール文章

2010/03/27


東京大学 近山・田浦研究室

山本 一成





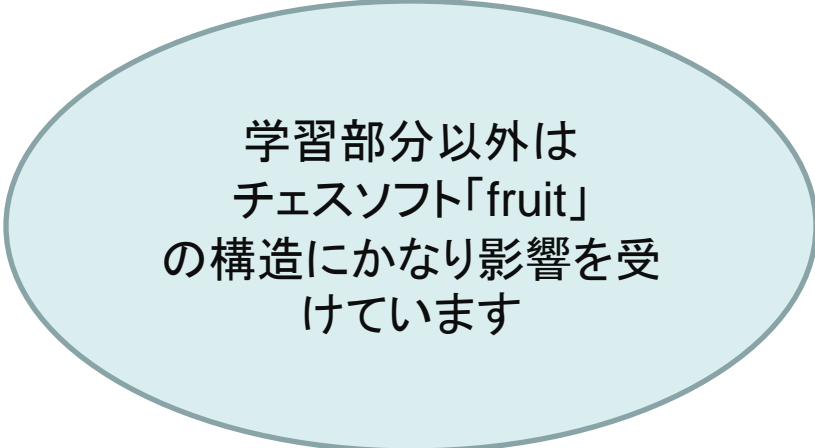
名前について

- Ponanzaってなんだよ
 - どう見てもBonanzaをもじっただけです
 - 本当にありがとうございました
 - 深い意味はなく、単純にギャグみたいに弱いのでこういう命名になりました(笑)
 - 保木さんの許可はとってあります
 - ライブラリのBonanzaは使用してません
- 



Ponanza基本構造

- 全幅探索(PVS,LMR)＋静止探索
- 詰将棋ルーチンは持っていない
- Bonanza Methodを使った評価関数の学習
- Bit Boardを使わず単純な配列 etc..

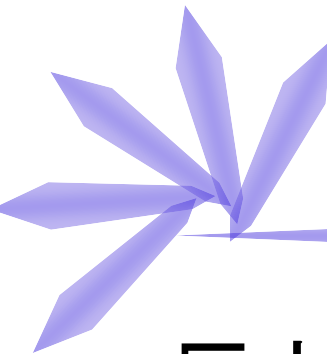


学習部分以外は
チェスソフト「fruit」
の構造にかなり影響を受
けています



これから先が本当のアピール

- アピール点は4つ
- 全体としてテーマは並列と簡潔




アピール1 (パラメタ数)

- 巨大なパラメタ数
- パラメタ総数は**4億個**
 - 実際は対称性を考慮して減ってはいるが
 - たぶん世界で一番多いはず

パラメタ総数は
Bonanzaのおよそ2倍
しかしBonanzaより強くはない
なんとか強くしたい



アピール2 (MapReduce)

- 膨大なパラメタをBonanzaMethodで学習するためには巨大な計算量が必要
 - 巨大な計算機環境をMapReduceモデルで学習を走らせることにより解決
 - 現在1000cpuを用いての学習に挑戦中
- 



アピール3 (最小の開発効率)

- 現在のプログラムの開発は1月ごろからスタートするも、ほぼ完成状態にある
- なぜならプログラム総行数わずか**3000行**
- 開発言語にC#を使用することでGC,ラムダ式,LINQ,コルーチンを容易に実現可能となり大幅な開発効率上昇となった




アピール4 (簡単な並列化)

- 従来のアルファベータ探索の並列化は非常に困難なものであった
- しかしTask Parallel Libraryを用いた並列化によりスレッドを明示的に使用せずアルファベータ探索を並列化することに成功しました

スレッドを明示的に使用しないことで開発効率を大幅に向上させることに成功



アピールのまとめ

- パラメタ数4億
 - MapReduceでBonanzaMethod
 - プログラム行数3000行
 - スレッドを明示的に使わない並列探索
- 

最後に宣伝(この場を借りて)

- 将棋専門誌 将棋世界に片上六段とコンピュータ将棋に関する話題で「コンピュータは七冠の夢を見るか？」という連載をしています。
- 興味がある方はぜひご覧下さい

こんな感じの雑誌です→

