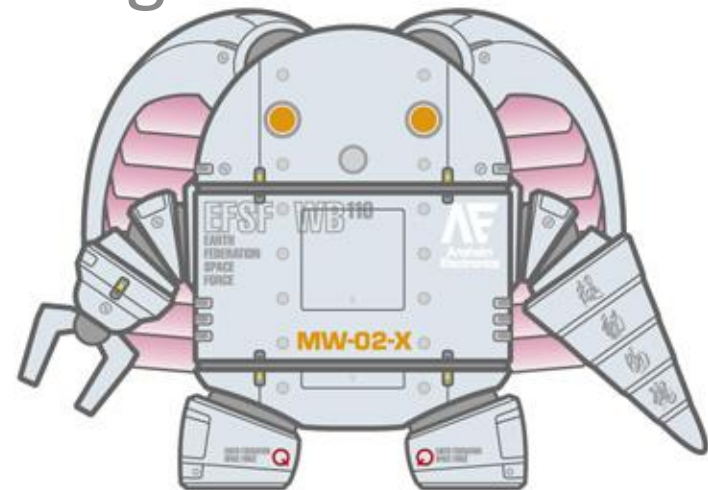


メカウーサー アピール文書

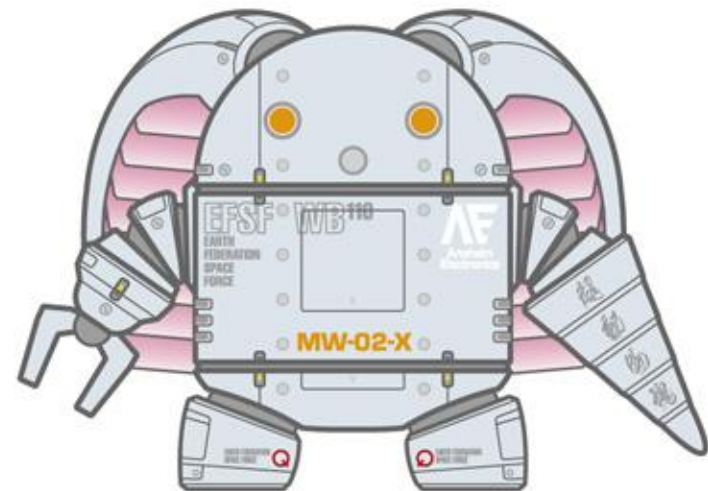
MechaWooser Technologies

木村 健



本文書について

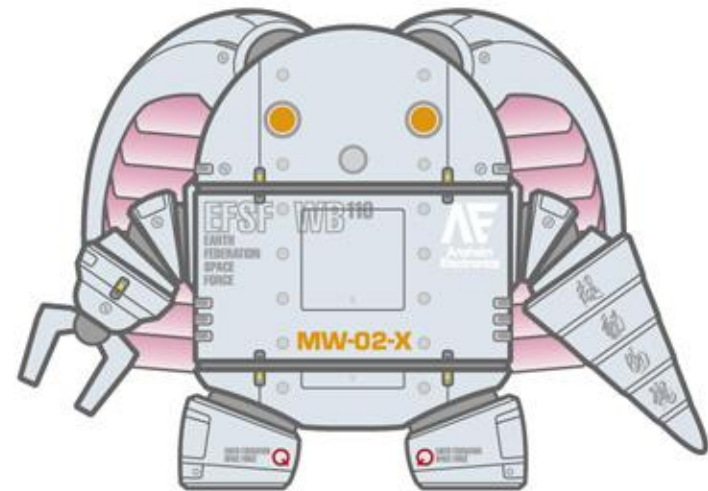
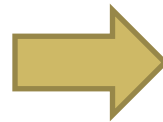
- 第20回コンピュータ将棋選手権における推奨事項であるアピール文書の提出を本PowerPointドキュメントの提出によりFulfillするものとする



そもそもメカウーサーとは

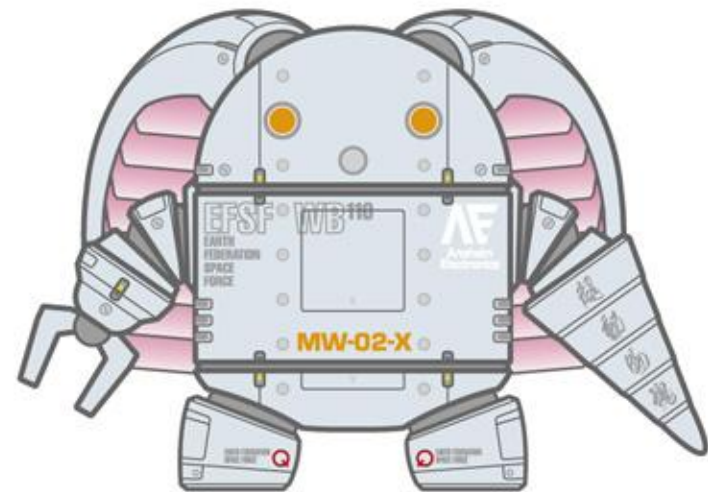
- うーサーさん (<http://twitter.com/wooser>) の創作した架空のモビルスーツ
- 転じてうーサーさんが特別に書き下ろした1MBに及ぶテキストデータより誕生したすーぱーtwitterぼっと(だぞ)

これがメカウーサー



よく間違えられるのですが・・・

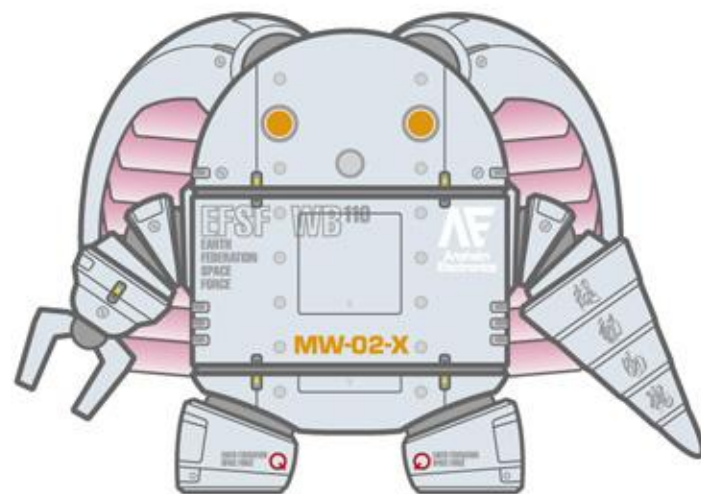
- メカウーサー (twitter bot) の作者は2名
- うーさー (wooser) さん
(文章担当、文章系サイト管理人)
- 木村 (kimrin)
(メカ=プログラム担当、
ただのPCヲタ)



なぜメカウーサーが将棋を？

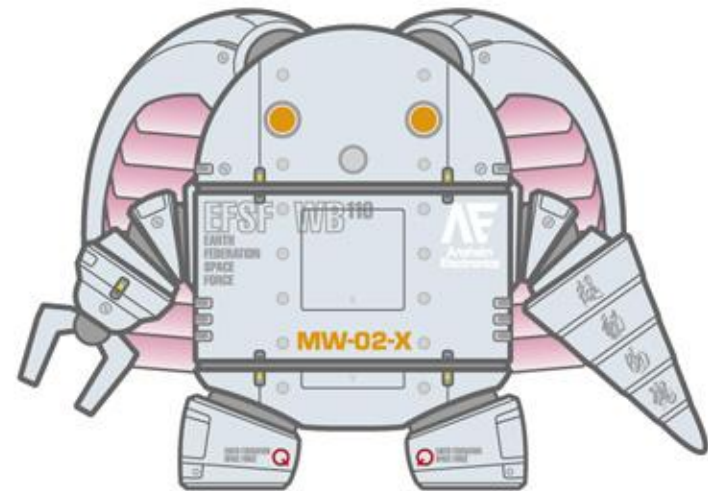
- 木村は実はCSAに所縁の深い農工大小谷研のOB(在籍当時は自然言語班に所属)
- Twitter botであるメカウーサーは小谷研当時に培った技術をもとに生まれた


→次に将棋プログラムを作るのは自然な流れ！



ところで…

- みなさんにとって、
萌えって
なんですか？

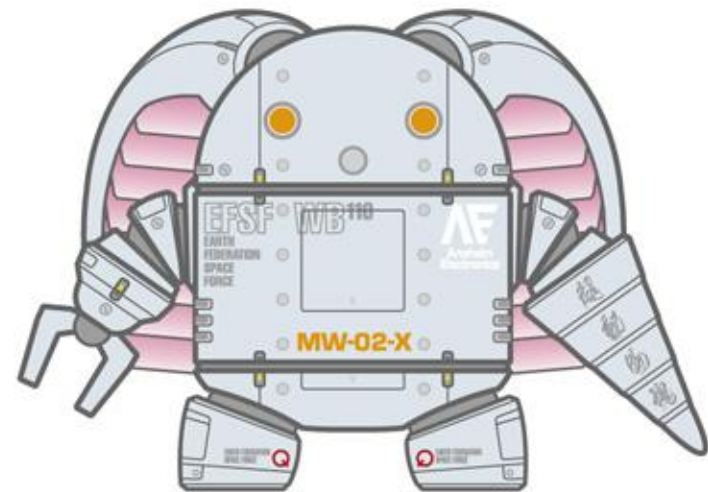


The background is a solid blue color. On the left side, there are several thick, overlapping lines in various colors: light blue, pink, orange, and yellow. These lines start from the left edge and move towards the right, with some curving and overlapping each other. On the right side, there is white Japanese text arranged in three lines.

おそれいりますが
しばらくそのまま
お待ちください

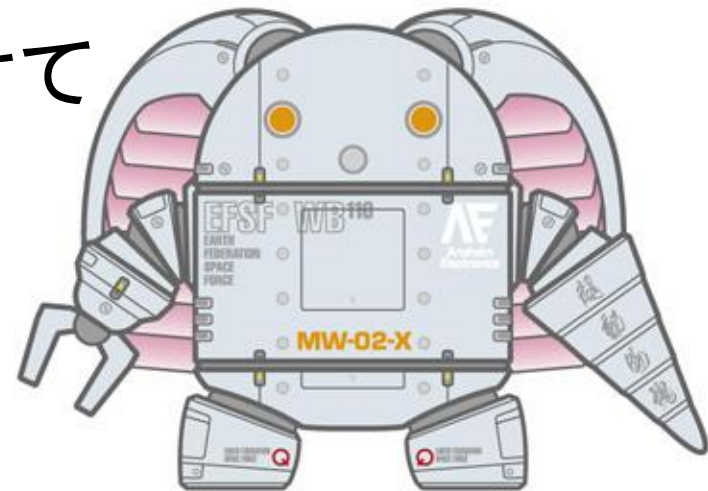
し、失礼しました

- スライドを間違えたみたいです。。。(_ _;)



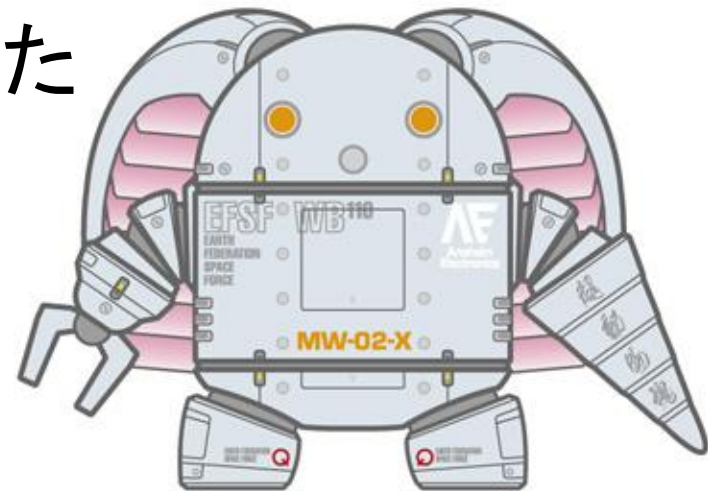
ここから本題

- 実験プロトタイプとしてのメカウーサー
 - 新しい技術を実験してみたい
 - → CUDA 技術の採用
- ひとまずLinux環境上で動くものを目指す
- 今年だけでなく、何年か続けてimproveしていきたい



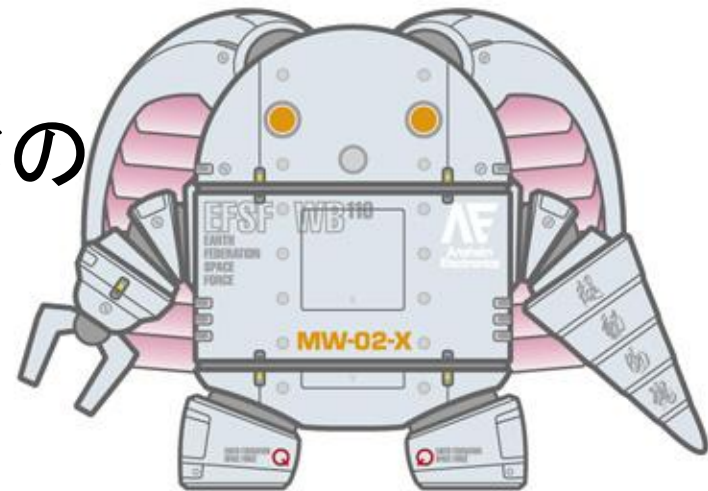
まじめな話ここから

- CUDA技術の採用
- 実は超並列環境がグラボいっこで実現してしまう
- しかし単純にX86で動いている将棋ソフトを移植しても、非力なソフトができるばかり
- そこでいくつか案を練ってみた



グラフィックボードについて

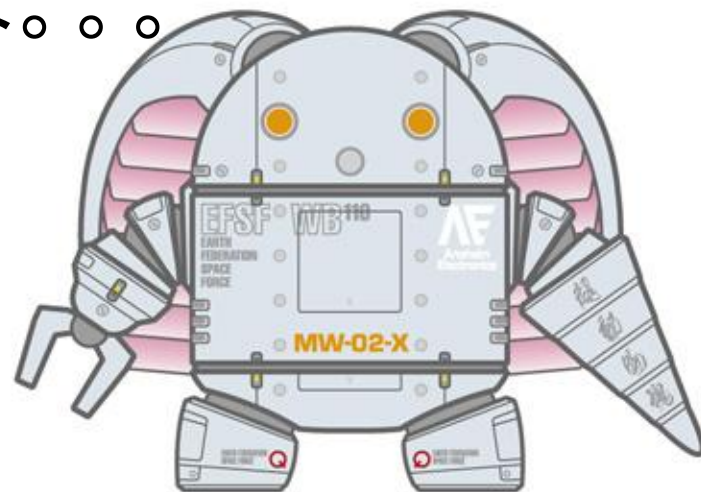
- GeForce GTX 285の場合
 - ストリームマルチプロセッサ数: 30
 - コア数: 240
- 同時に(仮想的にではありながら)数万のスレッドを同時実行することが可能
- メモリは1GB: ただしレジスタの100倍のアクセス時間



CUDA上での将棋プログラム

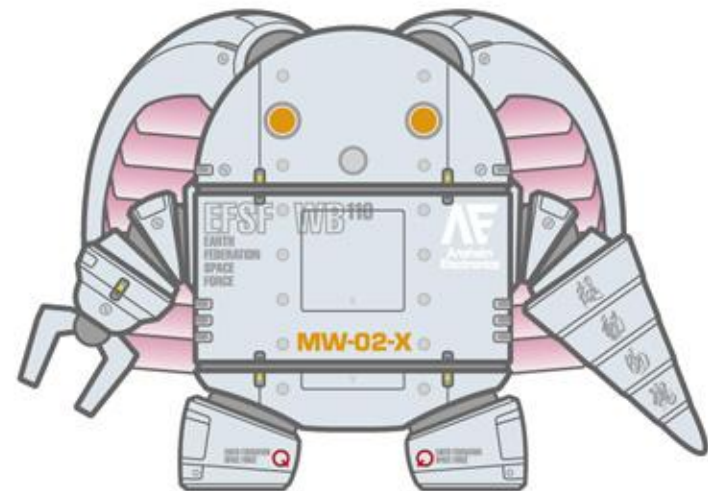
- たとえば全幅探索（キューを使って）
 - （卓上での）計算の結果、5手先の探索で
グラボのメモリ全体をキューとして使い
きってしまうことが判明 → 理論上の限界
- ではやっぱりまじめに、並列探索
勉強しないとできないのかな。。。

→ YES



CUDAによる並列探索

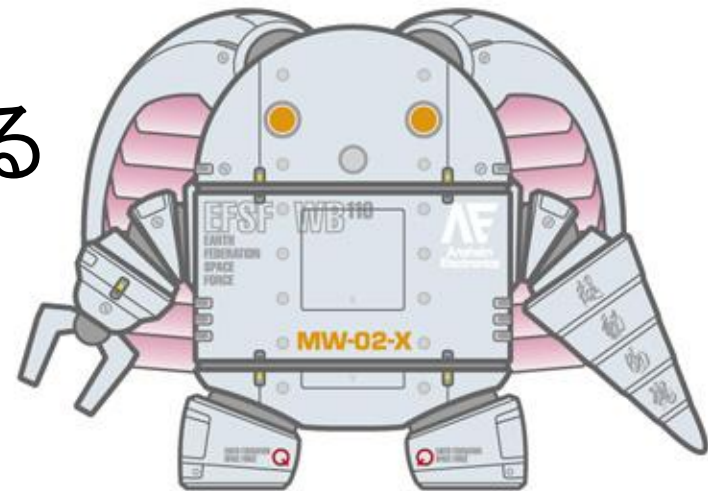
- とりあえずYBWCを参考に自分なりのアルゴリズムを実装(してみたい)
- YBWC: Young Brothers Wait Concept
1993年のFeldmannのドクター論文
- 要するに兄弟ノードの順序関係において、
弟(若いノード)を兄(年配のノード)が待つ(=同期する)



並列探索のアーキテクチャ

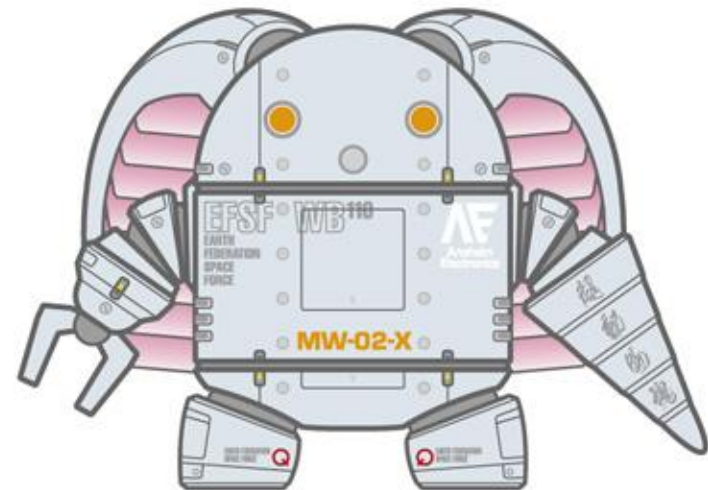
(ここからYBWCを参考にした自論)

- Managed threadsとWorking threadsという二種類の実行スレッドを用意する
- Managed threads は複数のWorking threadsを管理する(たとえば16スレッドに1スレッド)
- Managed threadは他のManaged threadから飛んできた仕事の依頼をWorking threadsに振り分ける



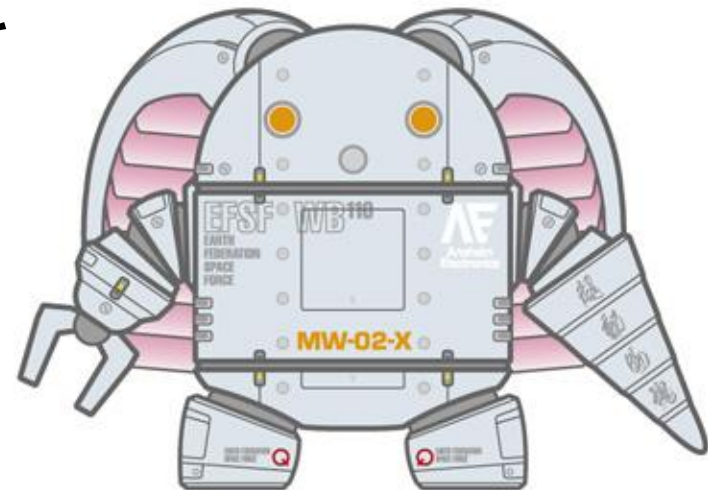
Working Thread

- Managed threadsから依頼を受けた仕事を実行する
 - 可能手の生成
 - 各子ノードの探索
 - Managed threadに仕事を依頼
 - Managed threadがthread numberをランダムに生成して、別のManaged threadに仕事を依頼する
 - 評価関数の計算
 - 結果をManaged thread経由で依頼元に返す



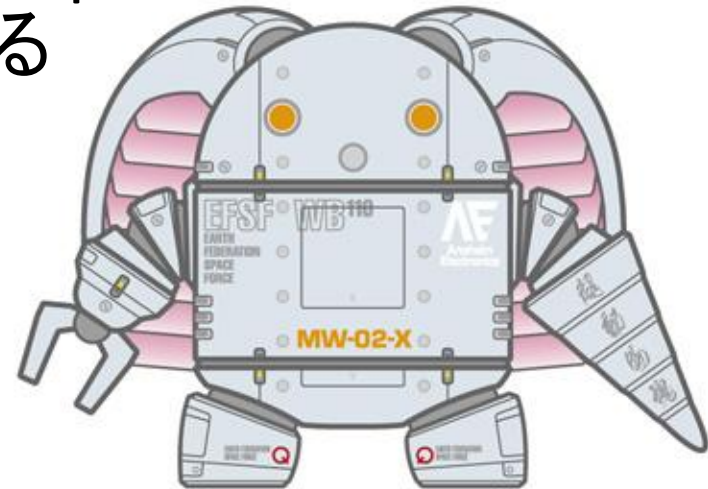
Managed threads(1)

- Managed threadは自分のキューを監視する
- 3種類の仕事がある
 - 他ノードからの探索の依頼の処理
 - 自分のmanageしているWorking threadsからの仕事の依頼の処理
 - Working threadsからの結果を依頼元に返す処理



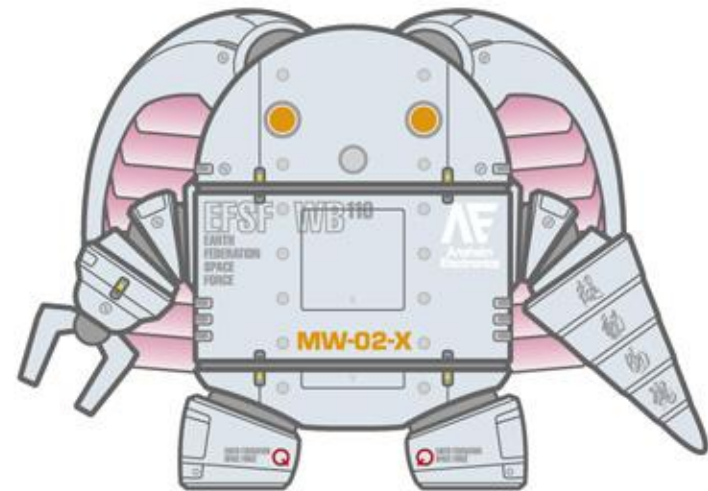
Managed threads(2)

- 他ノードからの仕事
 - 単純に自分の Working threadsのどれかに割り振る
- 自分のmanageしているWorking threadからの作業の依頼
 - 専用にキューを用意して、principalなノードから順に仕事を他ノードに依頼する
 - 専用のキューで同期をとる (結果が帰ってきてから次のノードの仕事を割り振る)



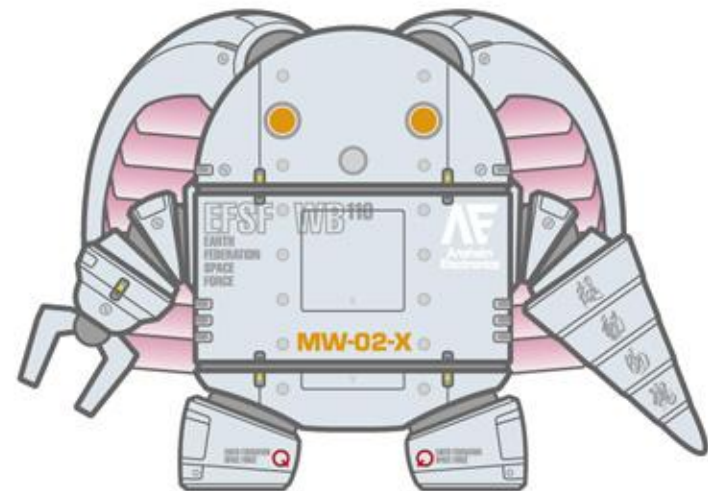
で、実は。。。

- 現在可能手の生成くらいまでできている
- Managed threads と Working threadsの実装はまだ。。。
- その他将棋プログラムに必要な機能もまだ手付かず ...orz



そういうわけで

- 今年の大会に出場できたら奇跡だと思ってください
- 来世^H^H来年会場でお会いできると、いいですね。。。
- なんとか5月に向けてひと踏ん張りしたいところです



以上です☆

- ご清聴ありがとうございました

