

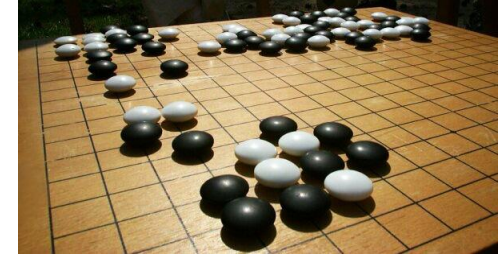
# 戦略的高性能計算システム開発に関する ワークショップ

美添 一樹

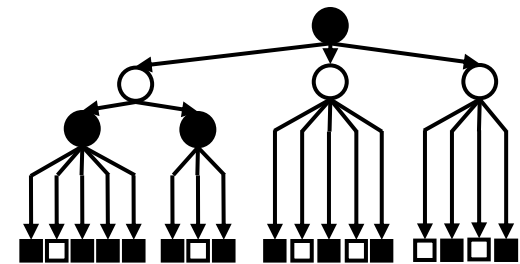
東京大学大学院 情報理工学系研究科

ポジション: アプリケーション

# 自己紹介



- 探索アルゴリズムの研究中
  - モンテカルロ木探索というアルゴリズムの並列化に取り組んでいます
    - 確率的な探索アルゴリズム
  - コンピュータ囲碁を強くしたい
  - 機械学習などにも興味あり



- 完全にアプリケーション側のポジション

# 今日のポジション(プロの開発者的な立場)

- 別に並列プログラミングは好きではない
  - 嫌いでもない
- アプリを書きます
  - パッケージ的なものも書く
  - 並列化しやすすくないアプリも書く
- アプリはたくさん書きます
  - 長期間一つのアプリをメンテしたりはしない
  - 一回書いたものは長期間動いてくれないと困ります
- でも速いことはうれしい
  
- 5年後よりはちょっと先の話かもしれませんが
  - 計算機センター限定の話でも無い
  - アカデミックっぽくない立場

# (希望) Write Once, Run Forever

- 時間の制約
  - アプリを開発しているときは時間がある
    - 技術もある程度ある
    - 多少使いにくかったり字面が難しくてもほとんど問題無い
- 希望
  - 多少実行環境が変わっても、メンテナンスフリーで動いて欲しい
    - ノード数が増えたとか通信が速くなったとかで仕事が発生すると困る
  - 再コンパイルしてもいい
    - ただし自動的にやって欲しい
    - (もしできればソースは非公開のまま)
- 全自動はもちろん難しい
  - アプリ開発者として、できる範囲で協力します

# アイデア

## 配布物 = プログラム + 最適化問題

- 最適化問題 (≡ 探索問題) のイメージ
  - パラメータのリスト
  - 評価基準 (実行時間、精度、速さなど)
- 性能に影響するパラメータをリストアップします
  - 環境が変わったら、自動的に良いパラメータを設定して欲しい
    - オフライン: 空時間にチューニング
    - オンライン: 実行するたびに更新
- セットで用意するので、後はよしなに
  - プログラム(バイナリ or ソース) + パラメータリスト

本質的には同じ

# パラメータの例

- つい最近作った並列探索プログラムのパラメータ
  - それぞれの意味はどうでも良いです
  - brute force なやり方にはかなり広い空間

探索空間として考えると、それなりに広いし、性質もバラバラ

double [0, 1.5] : ある数式の定数 c  
double [0, 100<) : FPUというパラメータ  
double [0, 1] : Virtual Loss parameter  
integer [0, 10<) : replace parameter 1  
integer [0, 10<) : replace parameter 2  
2択 (use rave, no rave)  
2択 (hash table method 1, method 2)  
2択 (通信をまとめる, まとめない)

⋮

# 探索空間の性質をヒントとして与える

- パラメータは多そうだが、自分には簡単な探索問題に見える
  - 性質の良さそうなパラメータが多い

## 評価基準

(速度、精度、強さ等)

入力によって変動するか  
確率的に変動するか

## 有望な箇所を指定

実数区間  $[0, 1]$  だが、0に  
近い方を重点的に

## 性質がきれいか

局所最適解に落ち込む  
心配が大きいかどうか

## 他のパラメータとの関係

これを変更すると  
パラメータX が大きく影響を受ける

# まとめ

- 用意するものは
  - プログラム(バイナリ or ソース) + 最適化問題
- 環境の変化に自動的に対応して欲しい
  - ノード数 / 通信速度 などが2桁くらい変わっても仕事をしたくない
- 機械学習や探索と連携できるのではないか
  - 一応、自分は探索が専門です