

2

あから 2010 のシステム設計と
操作概要

保木 邦仁¹ 金子 知適² 横山 大作²
小幡 拓弥¹ 山下 宏³

1 電気通信大学 2 東京大学 3 フリーランス

はじめに

あから 2010 (以下「あから」) の設計方針は、これまでの正統派ゲームプログラミングの道とは大きくかけ離れている。思考型ゲームにおける大規模人工プレイヤとして、チェスにおける IBM の Deep Blue がしばしば引き合いに出される¹⁾。1997 年にチェス世界チャンピオンの Garry Kasparov 氏に 2 勝 1 敗 3 引き分けで辛勝したこの人工プレイヤは、プロジェクト用に設計されたチェス専用プロセッサと、このハードウェア専用書き直されたチェスプログラムからなり、IBM 社の威信をかけ綿密にチューニングされた特別製であった。

一方、あからは既存計算機を可能なかぎりかき集め分散並列化を行い、さらに既存の複数将棋プログラムをそのまま利用して強いプレイヤを作る、という基本理念の下で作られた。これに沿った計画はたいへん欲張ったもので、激指、GPS 将棋、Bonanza、YSS^{☆1} を各作者が各自改良しつつ、それと並行してシステム全体の構成を行い、なおかつ実際に使用する計算機数および種類は当日になるまで確定させな

^{☆1} (激指) 毎日コミュニケーションズより市販されている将棋プログラム。作者は鶴岡慶雅、横山大作ほか。2010 年世界コンピュータ将棋選手権優勝プログラム。(GPS 将棋) <http://gps.tanaka.ecc.u-tokyo.ac.jp/gpsshogi/>、作者は金子知適、田中哲朗ほか。同 URL によりソースコードが公開されている。(Bonanza) マグノリアより市販されている将棋プログラム。保木邦仁作、思考部のソースコードは http://www.geocities.jp/bonanza_shogi/ より公開されている。(YSS) AI 将棋 Version 17 (e frontier, 2009) として市販されている将棋プログラム、山下宏作。

い、というものである。

このような一風変わった計画の成立は、ゲーム情報学の新技术である「合議法」と「ゲーム木分散探索法」により支えられていた。本稿では、これらの新技术を中心とし、並列化度を高くすることにより棋力が向上する仕組みおよび数十～数百台の計算機を繋げたシステムの安定化なども含めて、あからのシステム概要と技術的準備について紹介する。

システム概要

あからのシステム概要を図-1 に示す。各クライアント群 A～C は、与えられた局面に対して次の一手を思考することのみを集中的に行うユニットである。これらの思考プログラムには、激指等 4 種類の将棋プログラムを使用した。これら多数のコンピュータ将棋プログラムは①、②および④のサーバ下に接続されている。

図中①は対局サーバであり、接続されているプログラム間で絶え間なくマッチメイク・対局を行わせるものである。このサーバプログラムにはコンピュータ将棋対局場 Floodgate にて 2 年以上の運用実績があるフリーソフトウェア、shogi-server を用いた²⁾。ここに、合議サーバに加え A 群の将棋プログラムも常駐させ、24 時間体制で互いに練習対局を行わせた。練習対局の内容は Web にアップロードし、開発者全員が好きな時間にこれを確認できる

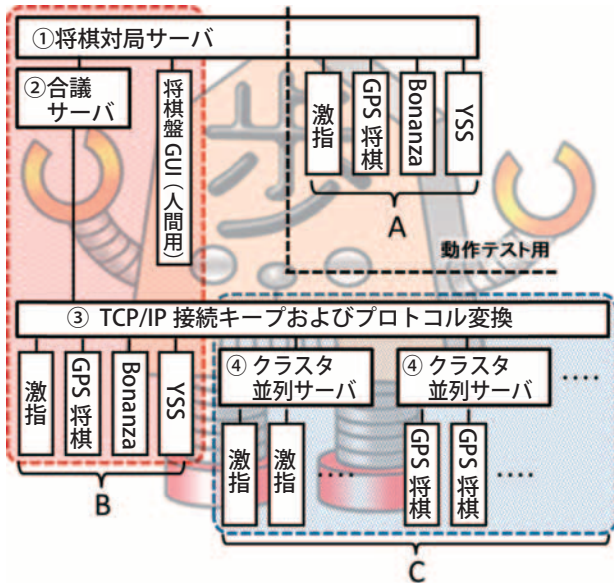


図-1 あから2010のシステム概要

ようになっていた。また、人間との対局は将棋盤 GUI を対局サーバに接続することにより行った。この GUI には高田淳一氏（現コンピュータ将棋協会，CSA 理事）作の SFICP に中断再開機能を組み込んだものを用いた³⁾。これも、長い運用実績のあるプログラムである。

図中②は合議サーバであり、ここより下の階層があからの思考の中核となる。合議サーバの大まかな役割は、クライアント間で意見を取りまとめて多数決により実際の指し手を決定することである。図中 B 群と C 群に示されるように、多数決に用いられるクライアントは複数存在し、これらのクライアントは対局中にも通信の接続・切断を自由に行うことが可能である。ただし、対局継続のためには最低1つのクライアントが接続されている必要がある。

図中③は合議サーバ各クライアント間で通信プロトコル変換を行うラッパーである。また、一定時間の通信不通等の異常が起きた場合の、クライアント再起動および再接続もこのユニットにより行われる。

図中青破線は、あからの必須構成要素を示している。通常の思考機能に加えて、相手思考中の予測読み機能は②合議サーバ、また中断対局再開機能は①対局サーバと②合議サーバならびに将棋盤 GUI により実現されていた。対局に用いる機能はすべてこ

の中に含まれている。これらの構成要素は数千局以上テストを経た信頼性の高いソフトウェアおよびハードウェアのみより構成した。

一方、図中青破線に示される構成要素は、できるだけ高い探索能力を得るため、ある程度の無理をすることが許されていた部分である。図中④はゲーム木分散探索を行うためのクラスタ並列サーバである。このプログラムはゲーム木の展開を独自に行い、その末端局面の評価を以下に接続の将棋プログラム群 C に手分けして行わせる。当日は、この並列サーバは計4つ起動され、激指等の4将棋プログラムをそれぞれの並列サーバに“できるだけ沢山”接続させた。

プログラム群 C のハードウェアは、東京大学情報理工学系研究科が所有するクラスタ istbs，科学研究費補助金「特定領域研究」“情報爆発時代に向けた新しい IT 基盤技術の研究”の支援により構築されたクラスタ InTrigger，ならびに各プログラムが所有する機材からなっていた。これは、異なるアーキテクチャとシステムソフトウェアが混在した異機種混合型環境である。また、このハードウェアは、istbs と InTrigger は東京大学本郷キャンパス，GPS 将棋所有のクラスタは東京大学駒場キャンパス，Bonanza 所有のクラスタは電気通信大学内と設置場所が異なり広域分散計算環境でもある（表-1 参照）。

プログラム群 C のハードウェアは、前述のように分散した多数の計算機の“寄せ集め”であり、対局中に通信途絶等のトラブルが発生することが予想された。一方、プログラム群 B はトラブルが許されない構成要素であり、このハードウェアの構成・耐久テストは慎重に行った。設置場所は、より安定した通信が期待できる対局室の隣である。また、設置台数も各プログラム1台にとどめて、信頼性の高い非分散並列版のプログラムを動作させた。使用した計算機は、事前に導入して動作検証が可能なことと、C 群のクラスタにトラブルが発生した場合にもある程度の探索性能が出せることを念頭におき、価格性能比が良好な以下の構成とした。

あから 2010 勝利への道

東京大学情報理工学系研究科 istbs クラスタ

Intel Xeon 2.80GHz, 2 cores	106 台
Intel Xeon 2.40GHz, 2 cores	60 台
小計	166 台 332 cores

激指, GPS 将棋, Bonanza が利用

科学研究費補助金「特定領域研究」情報爆発時代に向けた新しい IT 基盤技術の研究, InTrigger クラスタ

Intel X5560 2.80GHz, 8 cores	10 台
AMD Opteron 2380 2.4GHz, 8 cores	2 台
Intel E5410 2.33GHz, 8 cores	9 台
小計	21 台 168 cores

激指, GPS 将棋, Bonanza, YSS が利用

東京大学, GPS 将棋所有クラスタ

Intel X5570 2.93GHz, 8 cores	1 台
Intel X5470 3.33GHz, 8 cores	1 台
Intel X5365 3.00GHz, 8 cores	1 台
AMD Opteron 2376 2.3GHz, 8 cores	4 台
AMD Opteron 280 2.4GHz, 4 cores	1 台
Intel Core2 Quad Q6700 2.66GHz, 4 cores	1 台
小計	9 台 64 cores

GPS 将棋が利用

電気通信大学, Bonanza 所有クラスタ

Intel X5680 × 2, 12 cores	4 台
Intel W3680, 6 cores	1 台
Intel Core i7 980X, 6 cores	1 台
Intel W3440, 4 cores	1 台
小計	7 台 64 cores

Bonanza が利用

YSS 所有マシン

Intel i980X 3.3GHz, 6cores	1 台
----------------------------	-----

YSS が利用

表-1 クラスタ並列探索に使用したハードウェア

- CPU: Xeon W3680 3.33GHz 6cores
- Memory: 24GB (DDR3 UMB ECC 4GB × 6)
- 4 プログラムそれぞれに 1 台ずつ, 計 4 台

合議サーバの仕組み

■ 異種プログラムによる単純多数決合議の概要

合議法を初めて採用した将棋プログラムは 2009 年世界コンピュータ選手権の「文殊」である⁴⁾。このプログラムの仕組みは、Bonanza の評価関数に乱数を加えて複数クライアントを生成し、3 台の計算機により多数決をとるといったものであった。このような単純な仕組みながら、文殊は他の強豪相手に 3 位という好成績を収めている。これは本家の順位

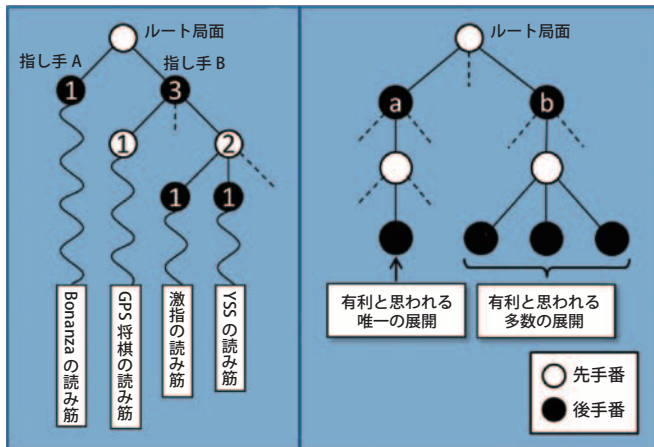


図-2 (左) 各プログラムの読み筋とその多数決の様子. (右) 多数決合議が有効に働くルート局面一例.

(4 位) より上であり、文殊の合議法が分散並列環境を有効に利用していたことを示唆している。

一方、あからが採用した合議法では、複数クライアント群を激指等の異種プログラムにより形成していた。この手法の仕組みを図-2 に示す。異なる 4 プログラムの読み筋は通常異なったものとなる。これらの手順は、初手から異なる場合や手順の途中から異なる場合等があり、たとえば図-2 左のような木構造を形成する。ノード内の数字は多数決の票数を表している、この場合は 3 票獲得した指し手 B が実際に指される。ここで、指し手 A よりも B の方が良いと主張する厳密な根拠はない。しかし、Bonanza のみが絶妙手 A を発見したという可能性よりも、激指等の他 3 プログラムに支持された B の方が良い手である可能性の方が高そうだという感触は得ることはできる。

このように、良い手である可能性が高そうな手を選択して、多数決合議が有効に働く局面の一例を図-2 右に示す。このルート局面で考え得る展開は 4 通りである。ここで、これら 4 つの局面の優劣は現在のいかなる高精度評価関数および探索手法では知り得ないとする。このような場合には、3 局面により支持される局面 b を選択するのが得策である。しかし、ある単一プログラムの思考結果のみを用いると、実際には局面 a がおよそ 1/4 程度の確率で選択されてしまう。この確率を下げるのが、複数



プレイヤー	勝率
4 プログラムの合議	73%
激指	50%
GPS 将棋	36%
Bonanza	62%
YSS	37%

表-2 基準プログラムに対する勝率
(共通の広い定跡データベースを利用)

プログラムの多数決により可能となるのである。

表-2 は、このプロジェクト用に行った多数決合議の実験結果である。初期局面からの指し手の変化を広めに設定した定跡データベースを用いて各プレイヤーが基準プログラム（激指）と1,000局対戦し、勝率を求めた。ここで、各プレイヤーが一手当たり消費する持ち時間は4秒程度である。この実験結果を見ると、各4プログラム単体での勝率は50%程度であるが、合議プレイヤーの勝率は70%を超えていることが分かる。激指等の4プログラムによる多数決合議は有効に働いていた。

さらに合議法は、分散並列環境を利用したプログラム性能向上という本来の目的に加えて、他の好ましい性質を持つ。まず1つ目は、前述の高度に分散並列化されたゲーム木探索システムの安定化である。将棋の対局では思考時間に制限が設けられていて、対局はリアルタイムで進行していく。そのため、数百台の計算機により形成される思考およびその内容出力の安定継続は、あからの満たすべき必須要件の1つであった。しかし、現在知られているゲーム木並列探索法は機器の故障に対する損害回避が難しい。これらの手法では、使用機器の数が10倍になるとトラブル発生率も10倍になるからである。また、フェイルセーフを重視すると探索効率が低下する。一方、合議法ではクライアントの通信接続・切断がいつでも可能である。合議法を採用したことにより、あからは分散並列処理の効率化と安定化という互いに相容れない性質を同時に満たすことになった。

合議法の2つ目の好ましい性質に、ある局面に対してどのプログラムの意見が多数派として採用さ

れるのか開発陣メンバでも見当がつかないことから、人間プレイヤーの特定プログラム対策を難しくするということが挙げられる。人間プレイヤーは対戦相手の長所・短所を分析して、自分にとってゲーム展開を有利に導くという素晴らしい能力を持つ。一方、将棋プログラムにはこのような対戦相手をモデル化して対策を講じるような機能はなく、できても対戦相手に応じた定跡手選択プログラミング程度である（本特集の第3編参照）。対局後日、対コンピュータ将棋対策を真剣に研究されている奈良女子大の篠田正人准教授（元アマ竜王）も「プログラムが4つあると対策に時間がかかる」とおっしゃっていた⁵⁾。異種プログラムによる合議が、人間の柔軟な相手適応能力への対策として一定の成果を上げていたのではないかと考えられる。

■ 合議サーバの動作内容

合議サーバは、トラブルを避けるために、極力単純な仕掛けで動作するように設計されていた。記述言語はPerlで約1,000行程度である。その動作内容は、大まかに以下の3つの状態に分けられていた。あからの手番が先手の場合には①からスタート、後手の場合には②からスタートする。また、中断局面からの再対局の場合には、中断局面までの手順を全クライアントに送信し、次の手番があからの場合には①、そうでない場合は②からスタートする。

① 自分の手番で指す手を思考する状態

ある一定の時間が経過したときに、各クライアントから集めた意見に対して多数決を取り、実際に指す手を決定。この手を全クライアントに送信して②へ。

② 相手の手を予測する状態

対局相手の指し手を受け取り次第、この手を全クライアントに送信して①に戻る。短めに設定された時間が経過した場合、各クライアントから集めた意見に対して多数決をとり、予測手を決定。この手を全クライアントに送信して③へ。

③ 予測した手に対する応手を思考する状態

対戦相手の指し手が届くまで、各クライアントの

意見を集計しながら待機。届いた手が予測手と一致した場合にはそのまま①へ、予測がはずれていた場合には訂正手を全クライアントに送信して①へ。

ここで、①中の“一定時間の値”は各クライアントの意見のバラつきを参考にして決定した。これにより、当然の一手は短時間で思考を打ち切り、有効な手が複数あるときには長時間思考することが可能となる。持ち時間（3時間）の当日の大まかな消費目標手数は、各クライアントの定跡データベースに登録されていない局面に到達してから、約60手であった。また、多数決を行う票の重みは、クライアントごとに変えられるように設計されていた。

同様に、合議サーバが期待する各クライアントの最低限の動作内容は以下ようになる。

I. 現局面を思考する状態

現局面を思考し続ける。最善手が更新され次第、これをサーバに送信。合議サーバからの通信を受信した場合IIへ。

II. 局面を遷移させる状態

合議サーバから指し手が届いた場合には、これに沿って現局面を更新。また、訂正手が届いた場合には、現局面を一手戻した後に訂正手に沿って局面を更新。Iへ戻る。

以上が合議サーバ・クライアントの基本動作となるが、このほかにもいくつかの補助的な通信内容をクライアントが受け付けて、必要な処理をしていた。

・キープアライブ

一定間隔で探索ノード数を通信して、各クライアントとその通信経路が活着していることを確認。

・序盤信号

クライアントが現局面を序盤と認識した場合にそのことを合議サーバに伝える。この情報は、たとえば2手目に△3四歩とするのか△8四歩とするのかで意見が割れて長考することを避けるため導入された。

・思考停止信号

クライアントの指し手がこれ以上時間をかけても

変化しない場合に、そのことを合議サーバに伝える。この信号は、クライアント側の定跡等のデータベース内に現局面を発見した場合や、最終盤にゲームが解けた（読み切った）場合に送信される。思考を停止したクライアントの意見のみで過半数の票を集めた場合に、合議サーバは次の手を即指す。

これらが、合議サーバに実装されていたほぼすべての機能となる。ほかにも、安定性を優先するため導入を見送った機能や、今回のイベントの経験・反省から生まれた新機能案も存在する。一例としては、長手数詰将棋プログラムの積極的利用が挙げられる。詰将棋を解くプログラムのアルゴリズムは、背尾による本格的な深さ優先証明数探索プログラムの開発以降急速に展開した⁶⁾。当然、通常の指し将棋は詰将棋とは異なったものであるが、それでも“100手以上の読み手順を短時間で正確に解く”という計算機の長所は、対する人間プレイヤーの作戦をある程度限定するには十分の威力を持つ。他の合議サーバ発展例としては

- 各クライアント固有の苦手パターン発見と投票荷重の変更
 - 次善手以下の投票とそれを利用した時間制御
 - 柔軟な予測読み
 - 世界コンピュータ将棋選手権で好成績を残し、なおかつプログラムの公開・市販を行っていないプログラムの追加参加
 - 成り、不成りのような意味が変わらない状況で意見が割れた場合は同一意見と認識
- 等が挙げられる。

クラスタを用いた並列探索の実現

多数決の枠組みが安定した後、筆者らは多数のコンピュータを協調動作させることによる強さの向上に取り組んだ。コンピュータ将棋の強さを支えるゲーム木探索アルゴリズムは、基本的には並列化可能ではないので、単純に計算機の台数を増やしても性能は向上しない。単純な並列実行では、逐次アルゴ

動作環境に対する頑健性

合議法と同様に、クラスタ並列探索においてもマスタ・スレーブ間は1秒に一度程度の通信で十分な性能を得ることが可能である(一般に、本格的な並列探索手法は、巨大な局面データベースの共有のため、ノード間の高速度な通信が要求される)。

クライアントに対する変更の少なさ

並列探索におけるスレーブの主要な機能は、新たな局面が割り当てられるたびに、ゲーム木探索を続けて最善手と評価値を報告することである。通常の将棋プログラムはすべて、この機能を持っている。また、マスタ・スレーブ間の通信プロトコルもGUI・思考エンジン間でよく用いられるUSI等を拡張したものとなっていた。新たに作成が必要だった機能としては、たとえば、将棋のルール上で許される指し手すべてを生成し(連続王手の千日手などの正確な反則検出も含む)有力な順に並べかえて指定の文法通りに報告する機能がある。

共通ソフトウェアの利用

並列探索を制御するマスタのコードはperlで書かれていて、ほぼすべてのOSやハードウェアで実行可能である。また、4プログラムで共通のコードを用いることにより、これを各プログラムにハードコードする手法と比べて、不具合改善を共通に行うことができるというメリットがある。

表-3 あから2010で用いられたクラスタ並列探索法の特徴

リズムでは可能となる効率化の多くを犠牲としてしまう。並列環境におけるゲーム木探索効率化のためには、これまでさまざまな設計や実装の工夫が提案されてきたが、さらなる研究の余地が残されている。

あからでは、実用的で頑丈な並列処理システムの構築が求められていたことから、合議システムと同様に、単純なマスタ・スレーブモデルが採用された。なお、本稿では混乱を避けるため、合議法に関する部分にはサーバとクライアント、クラスタ並列化に関する部分にはマスタとスレーブという言葉を用いる。動作の仕組みは以下のようである。

- マスタがルート局面を含む小さなゲーム木を構築し、各葉節点と内部節点以下の巨大なゲーム木の探索を各スレーブ(各計算機)に割り当てる。
- 葉節点を割り当てられた計算機はすべての指し手を、内部節点を割り当てられた計算機は子節点が作られていない指し手を担当する。
- 各スレーブは最善手や評価値が変わるたびにマスタに報告し、マスタはルート局面の最善手を更新して合議サーバに適宜報告する。

この方法が強さの向上に貢献する根拠は、マスタが構築するゲーム木が適切でさえあれば、ルート局

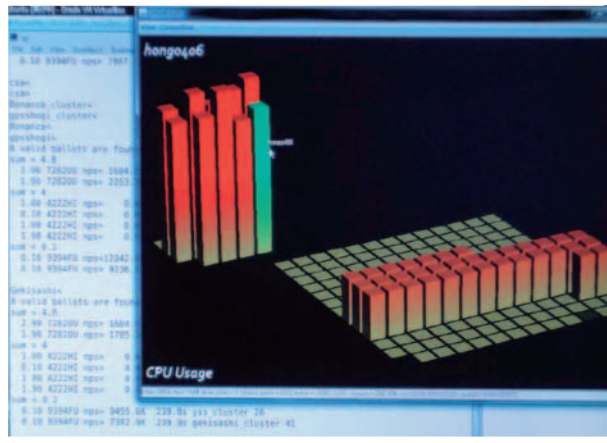


図-3 クラスタ(InTriggerとistbs)のリソース監視ツール

面よりも深い節点から探索したスレーブの探索結果が反映されるため、計算機1台によるルート局面の探索よりもさらに深い探索が実現されるという点にある⁷⁾。また、この手法は、表-3にまとめたように、比較的少ない変更で確実な効果を与える。限られた開発期間、少数の開発者、開発者自身がテストを担当という体制であったため、この手法がおそらく唯一の選択肢であった。マスタ部の記述言語はperlで約2,500行程度である。

さまざまな理由で、あからと等しい構成でのシステムの運用は本番でしかできなかったため、クラスタを用いた並列探索がどの程度の効果を上げたのかは不明である。しかし、GPS将棋を用いた一手15秒の対局試験では、8台の分散探索は1台に対して7割以上の勝率を残している。また、分散探索とメモリを共有する並列探索との比較でも、分散8並列とメモリ共有4並列、また分散16並列とメモリ共有8並列の勝率がほぼ拮抗しており、順調な性能向上となっている。このことから、あからにおいても十分な強さの向上があったと考えられる。図-3は、当日使用した機器のリソース使用状況の監視に用いたツールである。1本の棒グラフが1台の計算機に対応し、区域によって、激指、GPS将棋、Bonanza、YSSの利用機器が分かれている。

ゲーム木探索の並列化アルゴリズムは、メモリ共有型と分散メモリ型の環境では異なる。当然、分散メモリ型で動作する手法の方が並列化の効率が悪く、

あから 2010 勝利への道

安定性もより低いものとなる。クラスタ並列化システムの安定性は念入りにチェックされた。マルチコアプロセッサの1コアのみ使用する逐次探索版スレーブを40程度用いた擬似的な分散探索の対局試験は短めの持ち時間で多数行った。さらに、本番に近い100台以上の機器を利用してメモリ共有型と非共有型の並列化を組み合わせた思考時間2時間半の対局試験や、人間側が大長考するなど、稀にしか起こらない状況のソフトウェアテストも行った。

インターネットを介する分散メモリ型の並列探索を行う場合に問題になりやすい点は、通信不通への対応である。あからには通信不通になったスレーブの切り離し・再接続を全自動で行うような実装がされていない。機器の故障・トラブルの検出は人の手で行われていた。具体的には、オペレータが眺めているログに長時間応答しない計算機が表示された場合、通信不通の原因を探し、pingに応答がなかった等の場合には以下のようにシステムの再起動を行うなどである。

- マスタプログラムを止める。この時点で合議サーバの多数決からは離脱し、すべてのスレーブが自動停止する。
- マスタプログラムに登録されているスレーブ一覧から、問題のホストを削除する。
- マスタプログラムを起動する。この時点ですべてのスレーブが自動で起動される。各スレーブが起動の際に評価関数等の巨大なファイルをNFS経由で読むため、すべての起動には1分近くかかっていた。
- マスタプログラムは起動後合議サーバに自動で接続する。合議サーバからは、初手からの指し手が送られてくる。指し手を1つ受け取るたびに思考打ち切り・再思考を繰り返していたため、現在の対局が行われている局面に追いつくまで数分かかっていたようである。

GPS将棋所有クラスタ(表-1参照)のみを用いた試験では、このような手動の切り離しが必要となる状況は発生しなかった。しかし、対局数日前の練習対局においてこの切り離し作業が頻繁に必要とな

り開発者を悩ませた。原因としては、使用経験のまったくないハードウェアをいきなり使用したこと、使用機器の数が単純に増えたこと、マスタとスレーブ間の通信経路が長く複雑になったことなどが考えられる。今回はやむを得ずクラスタ並列システムの票の重みを低く設定して安全を期したが、ハードウェアトラブルや通信不通にも耐え得るシステムの設計は次回に向けての課題となっている。

おわりに

振り駒の結果「あから」が後手となった。後手の場合に準備していた戦法は4手目△3三角である(本特集の第3編参照)。この戦法の手順は図-1B群のYSSの定跡データベースに登録されていた。実際に登録した手順は、2手目△3四歩、4手目△3三角、6手目向飛車という方針のもと、下記9通りである。

1. ▲7六歩△3四歩▲2六歩△3三角▲6六歩△4二飛
2. ▲7六歩△3四歩▲2六歩△3三角▲4八銀△4二飛
3. ▲7六歩△3四歩▲2六歩△3三角▲6八玉△4二飛
4. ▲7六歩△3四歩▲2六歩△3三角▲7八金△4二飛
5. ▲7六歩△3四歩▲2六歩△3三角▲2五歩△2二飛
6. ▲2六歩△3四歩▲2五歩△3三角▲7六歩△2二飛
7. ▲2六歩△3四歩▲2五歩△3三角▲4八銀△2二飛
8. ▲2六歩△3四歩▲7六歩△3三角▲2五歩△2二飛
9. ▲2六歩△3四歩▲7六歩△3三角▲6六歩△4二飛

対局後、プロ棋士の佐藤康光9段や勝又清和6段に、「先手が7手目▲6八玉と指した場合、これを咎める居飛車を指せるのか」と指摘を受けた。しかし、このあから用定跡では上のように6手目までしか登録されていない。7手目以降の指し手は、各プログラム独自の定跡データベースの内容と、それらによる合議の結果に任せることになっていた。

専用定跡の内容を優先して合議サーバに選択させるために、はじめは票の重みが大きい定跡専用クライアントを合議サーバに接続し、この定跡を外れた



【コラム】 クラスタは卑怯？ ～クラスタ利用の背景～

計算機を何台も用いて、女性一人と相対するのは卑怯に見えるとの感想を持たれた方も少なくない。このテーマについてはさまざまな考え方があがるが、現時点ではできるかぎりの手段を尽くして最高性能のコンピュータ将棋を用意することが対局に臨むにあたっての礼儀であろうと筆者らは考えている。パーソナルコンピュータ上で動く将棋プログラムではトッププロには及ばない。ハードウェアや探索節点数制限などのレギュレーションによりコンピュータ側の強さに制限を課するのは時期尚早である。実際に、将棋連盟との打合せでも、最高の性能を実現するためにハードウェアを当日に変更することが認められていた。開発者の立場としてはハードウェア等の諸条件に特にこだわりはなく、とにかく将棋における人間対人工知能の歴史が棋譜として残ることを切に願っている。

ときに接続を切断する方法を取っていた。しかし、この方法と合議サーバの予測読み機能が上手に連系できなかったため、「この手は絶対に指せ」というクライアントの要求を合議サーバが呑むよう変更した。

あから用定跡の内容は開発メンバーの間でもあまり公にされておらず、あからそのものを用いた事前テストは、一部の人だけが極秘で“こっそり”4局行ったのみだった。そして、イベント前日になり、計算機100台程度使用したテストで専用定跡の動作確認を10局程度行った。

さて、大会当日30分前、約200台の計算機からなる本番構成のあからで初手から数手を2～3局テストすると、ときどき4手目に△3三角を指さないことが判明した。原因は、専用定跡担当のB群YSSの通信より早いタイミングで、他プログラムの思考停止信号が一定数合議サーバに集まり、多数決が決まってしまったことにある。なんとという初歩的なミス！本番対局の直前になり、合議サーバのプログラムコードを修正するか、しないか、開発者全員で合議をとった。結果、修正は行わずに通信タイミングおよび各プログラムの疑似乱数系列生成の成り行きに任せることとなった。実際に4手目△3三角が指せたことは、本当に運がよかった。

参考文献

- 1) Campbell, M., Joseph Hoane, J. A. and Hsu, F.: Deep Blue, Artificial Intelligence, 134, pp.57-83 (2002).
- 2) 森脇大悟, 金子知適: 自動対戦サーバFloodgate, コンピュータ将棋協会誌, Vol.20, pp.3-10 (2010).
<http://wdoor.c.u-tokyo.ac.jp/shogi/>
- 3) Shogi Framework Implements CSA Protocol, <http://www.junichi-takada.jp/sfcp/>

- 4) 伊藤毅志: 合議アルゴリズム「文殊」: 単純多数決で勝率を上げる新技術, <小特集> コンピュータ将棋の新しい波, 情報処理, Vol.50, No.9, pp.887-894 (Sep. 2009).
- 5) 篠田正人: トッププロ棋士との特別対局「清水市代女流王将 vs あから2010」を振り返る, イブニングセッション, 第15回ゲームプログラミングワークショップ (2010).
- 6) 小谷善行編著, 岸本章広, 柴原一友, 鈴木 豪共著: ゲーム計算メカニズムー将棋・囲碁・オセロ・チェスのプログラムはどう動くー, コンピュータ数学シリーズ7, コロナ社 (2010).
- 7) 金子知適, 田中哲朗: 最善手の予測に基づくゲーム木探索の分散並列実行, 第15回ゲームプログラミングワークショップ (Nov. 2010).

(平成22年12月9日受付)

保木邦仁 (正会員) kunihito.hoki@gmail.com

2003年東北大学理学研究科化学専攻修士(理学博士)。同年よりトロント大学化学科博士研究員。2006年より東北大学にて研究員、助手として教育研究に従事。2010年より電気通信大学特任助教。著書「ボナンザVS勝負脳ー最強将棋ソフトは人間を超えるか」(角川新書, 共著)。

金子知適 (正会員) kaneko@graco.c.u-tokyo.ac.jp

1997年東京大学教養学部卒業。2002年同大学院総合文化研究科博士課程修了。博士(学術)。同年同大学院総合文化研究科助手, 2007年助教。思考型ゲーム, 機械学習に興味を持つ。

横山大作 (正会員) yokoyama@tkl.iis.u-tokyo.ac.jp

2006年東京大学より博士号取得。博士(科学)。同大新領域創成科学研究科助手などを経て, 2009年より同大生産技術研究所助教, 現在に至る。並列プログラミング環境, ゲームプログラミングに関する研究に従事。

小幡拓弥 (学生会員) obata@minerva.cs.uec.ac.jp

2009年電気通信大学電気通信学部情報工学科卒業。同年より, 同大学院電気通信学研究科情報工学専攻博士前期課程入学, 在学中。2009年合議法を搭載したコンピュータ将棋プログラム「文殊」のメインプログラムとして世界コンピュータ将棋選手権に初出場3位。

山下 宏 yss@bd.mbn.or.jp

1995年東北大学工学部資源工学科卒業。(株)富士通を経てフリーランス。将棋, 囲碁ソフトの開発に従事。コンピュータ囲碁フォーラム理事。