

## ロビーサービスのシステムを用いた分散処理によるゲーム木の探索の高速化

郡司 直廉、森 秀樹、上原 稔

東洋大学工学部情報工学科

E-mail: n\_gunji@mo.cs.toyo.ac.jp, mori@cs.toyo.ac.jp, uehara@cs.toyo.ac.jp

本研究では、人工知能による二人零和有限確定完全情報ゲームのゲーム木探索処理を、分散処理することで高速化するシステムを提案する。本システムは、ロビーサービス型のゲームサーバ上において人間とコンピュータ（人工知能）が入り混じって対戦できるものであり、ゲーム木の探索処理をサーバに接続した複数のクライアント PC で分散処理するものである。人工知能専用のコンピュータに比べると、マシン 1 台あたりの処理能力は低いものとなるが、低コストで台数を多く集めることで高速化が実現できるという特徴がある。

### The speeding-up of Game-Tree Search for Lobby Service System by Distributed Computing

Naoyuki Gunji, Hideki Mori, Minoru Uehara

Dept. of Information and Computer Sciences, Toyo University

E-mail: n\_gunji@mo.cs.toyo.ac.jp, mori@cs.toyo.ac.jp, uehara@cs.toyo.ac.jp

In this research, we propose a system, which speeds up a game-tree search process in distributed computing with artificial intelligence techniques. The system is suitable for lobby service type game servers, where a person and a machine will fight each other. The game tree search process is performed with a lot of client machines connected to the game server. The power of each machine is lower than special purpose computers dedicated for the artificial intelligence. However, there is an advantage that high speed game tree search is realized using a collection of low cost machines.

#### 1. はじめに

コンピュータが登場して以来、人工知能の研究は盛んに行われている。その中でも、IBM の Deep Blue[3]が人間のチェスの世界チャンピオンであるカスパロフに勝ち、コンピュータがチェスの世界最強となったのは有名である。しか

し、将棋や囲碁の人工知能はまだ人間のほうが強く、例えば将棋の人工知能であれば、その強さは人間のプロの3段程度である。その理由は、将棋や囲碁はチェスよりも複雑だからである。たとえば、チェスは後半になるにつれて駒の数が減っていくのに対し、将棋は持ち駒制度があるために、常に駒の総数は一定である。そのた

め、チェスのように終盤をデータベース化しておき、それを用いて必勝手順の通りに手を進めるという方法を取りにくい。また、持ち駒は空いているマスならどこにでも置くことができるため、複数種類の持ち駒があると分岐因子が一気に増大する。そのため、チェスの分岐因子が平均 35 程度なのに対し、将棋では 100 前後になる場合もある。囲碁は盤面の広さが 19×19 もあり、そのどこにでも石を置くことができるので、分岐因子はさらに多くなる。

一般に、チェス、将棋、囲碁、リバーシ (オセロ) などの二人零和有限確定完全情報ゲームの人工知能は、ゲーム木という探索木を用いて先読みを行い、より自分に有利な局面になるように手を進めるという方法を取る。しかしゲーム木の探索には、先読みの手数や分岐因子が多いほど時間がかかる。普通は枝刈りアルゴリズムを用いて無駄な探索を省いたり、定石 (定跡) データベースを用いたりするので単純に先読みの手数と分岐因子だけでは比較することはできないが、探索速度はまだまだ向上させる必要がある。[1][2]

本研究では、このような問題点を解決するため、ロビーサービス型のゲームサーバと、それに接続した汎用のパーソナルコンピュータを用いて分散処理を行う方法を提案する。この方法の特徴は、インターネットに接続された汎用の PC を用いるという点と、ユーザーがゲームで遊んでいる間に CPU パワーを提供してもらうため、台数を安価に集めやすいということである。また、PC を提供するユーザーなら誰でも高度な人工知能と対戦できるという特徴もある。

本論文では、提案システムの設計と実装について議論し、実験を行うことで提案システムの有効性を確認する。

## 2. 関連研究

IBM の Deep Blue は、木の探索の高速化に並列処理を用いている。Deep Blue は POWER2 Super Chip(P2SC)を搭載したノードを 32 持つ並列コンピュータに、合計 512 個のチェス専用 VLSI プロセッサを搭載した、チェス専用コンピュータである。このようなコンピュータは非常に強力であるが高価であり、一般人が気軽に人工知能と対戦するというようなことは不可能である。

また、低価格で高い計算能力を得る方法として、グリッド・コンピューティングというものがある。これは、ネットワークで接続されたコンピュータの資源 (アイドル時の CPU パワーなど) を利用してサーバーから送信したデータの解析などを行うもので、現在、distributed.net[4]や、SETI@home[5]といったプロジェクトが行われている。この方法は、低コストで高い処理能力を得られるが、リアルタイム性に欠けるためゲームの処理には使いにくい。また、CPU パワーを提供したユーザーが得られる恩恵が無い (あるいは、得られる確率が非常に低い) ため、プロジェクトにそれなりに興味のある人からしか CPU パワーを提供してもらえない、という問題点がある。

## 3. 人工知能の仕様

本研究は、分散処理によるゲーム木の探索の高速化を目的としたものであり、ゲームには比較的単純なリバーシを用いた。リバーシは、以下のような理由でアルゴリズムの実装が容易であるからである。

- 必ず 60 手で終わる
- 分岐因子が少ない(特に終盤になるほど置ける場所が減る)
- 評価関数が単純

ゲーム木の探索を高速化するためには、分散化の他に、枝刈りや評価関数のアルゴリズムの最適化も重要である。しかし、これらは各ゲームに固有のものもあるため、最適化は各ゲーム毎に行う必要がある。そのため本研究では、各ゲームに依存する部分に関しては、人工知能の基本的なアルゴリズムとした。具体的には以下の通りである。

- ゲーム木の探索法は、一般的なミニマックス法を用いた深さ優先探索とした。
- 枝刈りのアルゴリズムには、各ゲームに依存しないアルファ・ベータ法を使用した。
- 評価関数は、リバーシの「相手に取られない石(角に置いた石など)の数が多いほど有利になる」という特徴から、そのような石に重みをつけて計算するアルゴリズムとした。

## 4. ロビーサービス型のゲームサーバのアーキテクチャ

### 4.1. 既存のサーバの特徴

ロビーサービスとは、サイト上で登録した ID などログインし、ゲーム会場となるサイト(ロビー)において、同じゲームを行っているユーザーなど、対戦相手を探すためのサーバである。

このようなサーバにログインしているユー

ザーの PC は、一般に、

- ゲームプログラムの実行
- ロビー画面の表示
- サーバとの通信

などの処理を行っている。ここで、ゲーム開始前や終了後にロビー画面を表示している状態では、(ユーザーがバックグラウンドで他の処理を行っていることも考えられるが)ユーザーの PC の CPU パワーは余っていることが多い。また、ゲームの種類が、チェスや将棋、またはカードゲームなどのような、処理内容が単純な画面表示を行うだけのものである場合、ゲーム中も CPU パワーが余っていることになる。結果として、サーバにログインしている間、常に CPU パワーが余っている可能性が大きいことになる。

### 4.2. 提案システム

本システムでは、ゲームサーバを、ユーザー同士の対戦の他にユーザー対コンピュータ(人工知能)の対戦も行えるようにする。この人工知能の処理を、ユーザーの PC の余った CPU パワーを利用して分散処理を行うことで、高度な人工知能との対戦を実現する。

図 1 は、ゲームサーバとクライアントの接続を図示したものである。既存のサーバと異なる点は、各クライアント PC が人工知能の処理を行う専用のスレッド(人工知能ノード)を持ち、人工知能との対戦を行うクライアントはさらに、分散処理の中心となる人工知能サーバのスレッドを作成するという点である。これらのスレッドはバックグラウンドで動作するため、プレイヤーは意識することなく CPU パワーを提

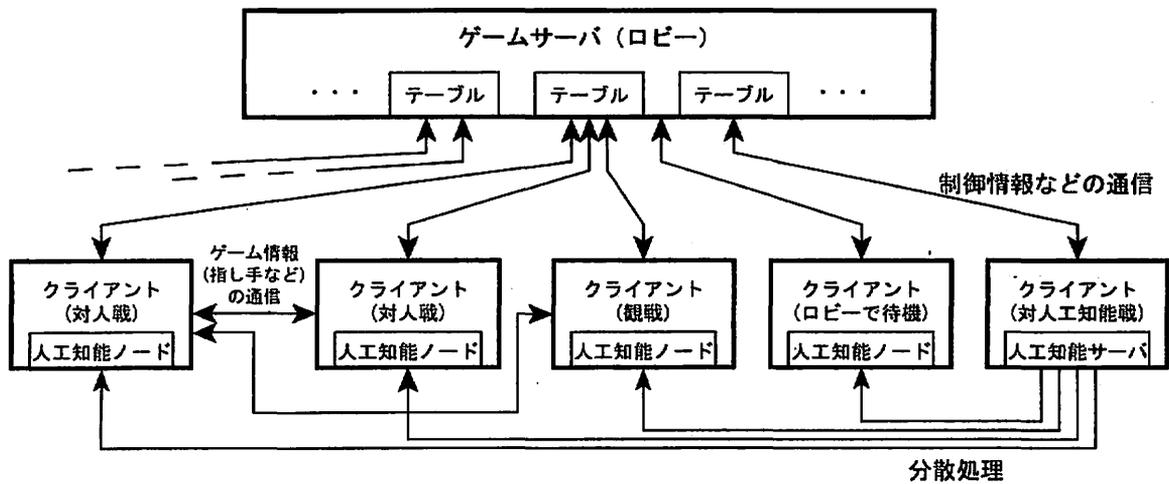


図1: 接続図

供できる。人工知能との対戦を行うクライアントが現れると、次の手順で分散処理を開始する。

1. 人工知能との対戦を行うクライアントは、人工知能サーバのスレッドを作成する。
2. ゲームサーバに対し、ノードの接続要求を出す。
3. ゲームサーバは、人工知能ノードをもつクライアントへ人工知能サーバのIPを送る。
4. IPを受け取ったノードは人工知能サーバへ接続し、分散処理を開始する。

1. まず、人工知能サーバは1手または2手先までを探索し、そこから先の探索を複数のノードに割り当てる。
2. ノードはサーバから割り当てられた節点から先を探索し、得られた評価値をサーバに返す。
3. サーバはノードから得られた評価値を元に、最適な手を選ぶ。

#### 4.3. 分散処理における探索アルゴリズム

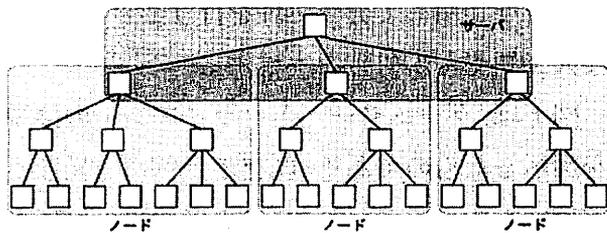


図2: 分散処理の手法

ゲーム木の探索において、木の根に近い部分は人工知能サーバが、そこから先は複数のノードが分散して探索を行う。図2にその模式図を示す。処理の手順は次のようなものである。

この方法の特徴は、通信するデータのサイズが小さく（現在は最大で21バイト、ただしパケットのヘッダ部を除く）、また通信の回数が少ないということが特徴である。また、1つのノードに対し、探索範囲の割り当てを2箇所分ずつ行うことで、1つ目の探索範囲が終了した後、サーバからの返答を待たずに次の探索範囲の探索を開始できる。そのため、最初の探索範囲の割り当て時と、最後の探索範囲の探索終了時以外はネットワークの速度の影響をほとんど受けることが無い。このため、低速なネットワーク環境でもその影響を受けにくい。

しかし、分散処理をすることにより、本来ならば枝刈りされる節点も一部ではあるが探索を行うため、その分だけ効率が悪くなるという欠点がある。

## 5. 実験と評価

### 5.1. 実験環境

速度差の大きい PC5 台を用意し実験を行った。使用した PC の性能を表 1 に示す。

なお、インターネットを経由した低速の環境での実験ができなかったため、10base-T LAN (リピータハブ使用) によってすべての PC を接続した。

	CPU	探索速度 (局面/秒)
PC1	Pentium 120MHz	2776.56
PC2	MMX Pentium 166MHz	4630.13
PC3	Pentium II 233MHz	8686.82
PC4	Pentium II 266MHz	9678.71
PC5	Duron 700MHz	22429.59

表 1: 実験に用いた PC の性能

各 PC の探索速度の測定には、分散処理を行わない、単一 CPU 用のリバーシの人工知能のプログラムを使用した。このプログラムでいくつかの局面に対して 7 手の先読みを行ったときの、

$$\text{探索した局面数} / \text{探索時間}$$

の平均値を探索速度とした。

### 5.2. 実験の内容

5 台の PC がゲームサーバに接続し、うち 4 台が対人戦、残りの 1 台が対人工知能戦を行っている場合を想定した。この場合、5 台の PC すべてが 1 つの人工知能の処理を分散して行

うことになる。なお、ここでは、人工知能サーバを動かす PC で、人工知能ノードのスレッドも同時に動作させている。その理由は、人工知能サーバは CPU パワーをほとんど必要としないため、サーバだけを動作させると CPU パワーが余ってしまうからである。

実験は、

- 序盤 (4 手後の局面)
- 序～中盤 (16 手後の局面)
- 中盤 (28 手後の局面)
- 中～終盤 (40 手後の局面)
- 終盤 (52 手後の局面)

の 5 つの時点からの 7 手先読みと、48 手指した後の状態からの全幅探索 (枝刈り無し、12 手 (つまりゲーム終了時点まで) 先読み) について、それぞれ複数回行い、探索にかかった時間を計測した。

また、以下の式より探索時間の理論値と分散処理の効率を計算した。

$$\text{理論値} = \frac{\text{探索した局面数}}{\text{PC1} \sim \text{PC5 の探索速度の合計}}$$

$$\text{効率} = \frac{\text{探索時間の理論値}}{\text{探索時間の実効値}} \times 100$$

※全幅探索は枝刈りを行わないため、探索速度が 5.1 で計測した値とは異なる。そのため、実験と同じ条件で各 PC の探索速度を計測し、そのデータから理論値を計算した。

### 5.3. 結果

表 2 に実験の結果を示す。

状況	探索した局面数	PC5のみ	PC1~PC5 分散処理		
		探索時間(秒)	探索時間(秒)		分散処理の効率(%)
			実効値	理論値	
4手後から7手先読み	47321.5	2.073	1.387	0.982	70.78
16手後 //	185941.5	7.697	4.752	3.858	81.18
28手後 //	355497.0	15.202	8.393	7.375	87.87
40手後 //	151596.0	6.545	6.415	3.145	49.03
52手後 //	485.0	0.025	0.060	0.010	16.77
48手後からの全幅探索	3854577.5	31.435	23.083	15.244	66.04

表 2: 実験の結果

速度差の大きな PC を使用していながら、ほとんどの状況において 5 台中最速の PC が単体で探索を行ったときよりも高速に探索を行うことが出来た。

分散処理の効率は平均で 70%程度であったが、ばらつきが大きく現れた。これは、使用した PC の速度差が大きかったのが原因の一つだと考えられる。速度差が大きい場合、探索に時間がかかる場所に高速な PC が割り当てられれば効率が良いが、逆にそういった場所に低速な PC が割り当てられると無駄な探索が増えて探索時間も増えるということが起こり得る。

52 手後の状況の 16.77%という値は、石を置ける場所が非常に少なかった(平均で 5 個所程度)のために無駄な探索が多く行われたということと、探索時間が平均 60ms と非常に短いために分散処理によるオーバーヘッドが大きく現れたことが原因である。

## 6. むすび

本論文では、ゲーム木の探索の高速化の手法として、ロビーサービス型のゲームサーバとそれに接続したクライアント PC を用いる方法を

述べた。また、実験を通じ、提案システムの有効性を検証した。

今後の課題としては、現状では人工知能サーバにトラフィックが集中する形となっているため、分散の段数を増やすノード(人工知能サーバから割り当てられた探索範囲をすべて探索するのではなく、2手先程度まで探索し、その先をさらに複数のノードに分散させる)を実装することを検討している。また、ゲームサーバが人工知能サーバに対して動的にノードを割り当てる方法、PC の速度差が大きい場合でもある程度の効率を保つ方法などについても研究を進める。

## 参考文献

- [1] 馬場口 登, 山田 誠二:  
人工知能の基礎, 昭晃堂(1999)
- [2] 松原 仁: 将棋とコンピュータ,  
共立出版株式会社(1994)
- [3] IBM: Kasparov vs. Deep Blue,  
<http://www.research.ibm.com/deepblue/>
- [4] distributed.net: <http://www.distributed.net/>
- [5] SETI@Home: <http://setiathome.ssl.berkeley.edu/>