

時間制約下での WWW 検索のための モバイルエージェント制御方法

國分俊介[†] 灰原清太郎[†] 川越恭二^{††}

指定時間内に処理を終了するという時間制約下で、モバイルエージェントに WWW 検索を代行させるとき、利用者の要求とモバイルエージェントの置かれている状況に応じたエージェントの行動を制御することが必要となる。そこで、本研究ではエージェントの置かれている状況を数値化し、その数値を用いて検索作業を優先させる度合いを示すエージェント優先度を定義する。さらに、エージェント優先度を使用し、各サーバの負荷度を考慮したエージェントの行動制御方法を提案する。実験の結果、負荷の大きいサーバでは検索せず、なるべく負荷の小さなサーバから検索することが可能となり、時間制約下で検索作業を行う回数を増加させるという結果が得られた。

Mobile Agent Control Mechanism for Web Retrieval under Temporal Constraint Environment

SHUNSUKE KOKUBU,[†] SEITARO HAIBARA[†] and KYOJI KAWAGOE^{††}

A control mechanism for mobile agent under temporal constraint environment is introduced. With the mechanism, mobile agents can determine their activity by themselves when the agent moves among several servers to do Web retrievals. The mechanism employs a special decision function of agent activity priority and server load. Some experimental results show that the number of Web servers where the agent can do retrieval increases, compared with the case where the agent moves sequentially.

1. はじめに

近年、携帯端末を用いたインターネット利用が急激に増加している。このとき、利用者は常時ネットワーク接続をするのではなく、通常、必要なときにネットワーク接続を行う。しかも、接続中には、利用者は短時間で必要な情報を入手する。本研究は、このようなモバイル環境で利用者が WWW 検索を行う場合を想定し、モバイルエージェントにより WWW 検索を代行させることを考える。このとき、モバイルエージェントがあらかじめ与えられたサーバを移動しながら利用者の要求を代行し、利用者が事前に指定した時刻、すなわち、利用者が次に接続する時刻までに検索作業

を終了しておかなければならないという問題を対象とする。

本論文では、あらかじめ指定した時間内でモバイルエージェントが戻ってくるように、エージェントが作業状況を考慮しながら移動できるようにする行動制御方法を提案する。

2. モバイルエージェントによる WWW 検索

2.1 概要

モバイルエージェントによる WWW 検索システムは、モバイルエージェントにあらかじめ要求、条件などを与え、利用者の代わりに WWW 検索を行うシステムであり、すでにいくつかの研究が進められている^{1)~3)}。その構成を図 1 に示す。

本論文では、利用者がエージェントサーバに要求を渡した後に生成されるエージェントが、あらかじめ指定された WWW サーバを巡回しながら、要求に従って情報を検索するものとする。なお、その間、利用者はネットワークから切断してもかまわない。

[†] 立命館大学大学院理工学研究科
Graduate School of Science and Engineering,
Ritsumeikan University

^{††} 立命館大学理工学部
Faculty of Science and Engineering, Ritsumeikan
University

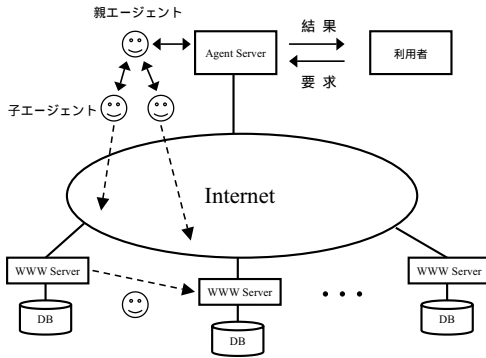


図1 モバイルエージェントによるWWW検索
Fig. 1 Mobile agent Web retrieval system.

2.2 時間制約下での問題点と解決方法

2.1 節のWWW検索システムにおいて、モバイルエージェントがデータを持ったまま巡回すると、WWWサーバやネットワークに負荷がかかる。その負荷が影響して、検索ができなかったり、利用者がネットワークへ再接続するまでに結果を受け取れなかったりすることが考えられる。そこで、時間制約下で利用者に検索結果を伝えかつ効率良い検索を行うために、モバイルエージェントの置かれている状況に応じた行動制御方法を提案する。3章で、エージェントの置かれている状況として累積検索処理時間、残存移動サーバ数、検索結果を含めたエージェントサイズを取り上げ、これらの状況とサーバの負荷度を考慮した行動制御方法を説明する。

3. エージェント優先度による行動制御

3.1 目的

2.2 節で述べた問題点を解決するために、モバイルエージェントの置かれている状況と移動先サーバの負荷に応じて、

- (1) そのサーバで検索する
- (2) そのサーバは後回しにする
- (3) エージェントサーバのもとへ戻る

の3種類の行動の中から状況に応じて適切な行動を選択するエージェントの行動制御を考える。その行動を決定する指標として、各モバイルエージェントに以下に示すエージェント優先度を設定する。

3.2 エージェント優先度

エージェント優先度 P は、エージェントがWWW検索において、行動を左右されると考えられる以下の3種類の要素から算出する。

- 現在までの検索時間の割合 P_{ret}
- 残り移動サーバ数の割合 P_{mig}
- エージェントサイズの割合 P_{agt}

その計算式を次式に示す。

$$P = s \times P_{ret} + t \times P_{mig} + u \times P_{agt}$$

ただし、1 エージェントが持てるデータ量はあらかじめ定められているものとする。また、 $0 \leq s, t, u \leq 1$ が成立する係数 s, t, u による重み付けにより、サーバの負荷の度合いによって、どの要素を重点的に考えるかを決めることができる。エージェント優先度は、各移動先サーバに到着時に算出される。

以降、各要素について説明する。

まず、時間制約下において、モバイルエージェントは設定された時間内で戻ってこなければならない。現在までの検索時間の割合 P_{ret} は、その時間の経過具合を測る要素である。 P_{ret} は以下の式で与える。

$$P_{ret} = \text{累積時間} \div \text{許容検索時間}$$

次に、時間制約下において、モバイルエージェントが複数のサーバを巡回しなければならないとき、なるべく多くのサーバを巡回したい。残り移動サーバ数の割合 P_{mig} は、現在のモバイルエージェントの巡回具合を測る要素である。 P_{mig} は以下の式で与える。

$$P_{mig} = \text{残存移動サーバ数} \div \text{総移動サーバ数}$$

最後に、モバイルエージェントは、移動と検索を繰り返すうちに、そのサイズが大きくなることが予想される。現在のエージェントサイズの割合 P_{agt} は、モバイルエージェントがどれだけデータを持っているかを測る要素である。 P_{agt} は以下の式で与える。

$$P_{agt} = \text{現在のエージェントサイズ} \div \text{許容エージェントサイズ}$$

3.3 行動制御方法

具体的に、エージェント優先度を用いたモバイルエージェントの行動制御方法を説明する。

まず、移動先のサーバにはサーバ負荷度 L ($0.0 \sim 1.0$) があるものとする。サーバ負荷度 L として、たとえば、CPU 使用率やサーバ内のエージェントの割合などがあげられる。移動先のサーバ負荷度 L とエージェント優先度 P を考慮して、以下のようにモバイルエージェントの行動を制御する。

- (1) エージェントサーバで生成されたモバイルエージェントは、移動先サーバリストの順に従って

表 1 サーバ負荷度によるエージェント優先度
Table 1 Agent priority with server load.

サーバ負荷大 ($L = 0.7 \sim 1.0$)	エージェントサイズと検索時間を重点的に ($s, u \gg t$) 考えて, P を算出する
サーバ負荷中 ($L = 0.3 \sim 0.7$)	すべての要素を平均的に考えて ($s = t = u$), P を算出する
サーバ負荷小 ($L = 0.0 \sim 0.3$)	常に検索するように $P = 1$ とする

移動を開始する.

- (2) モバイルエージェントは, サーバに到着したら時間とその時点のサーバ負荷度 L を測定し, そのサーバにおけるエージェント優先度 P を計算する. ここで, 表 1 に示す方法を用いて, エージェント優先度を算出する.
- (3) エージェント優先度 P とサーバ負荷度 L に関して, 判断関数 f を用いて, $f(P, L) \geq 0$ となるときは検索をし, それ以外は検索をしない. 判断関数 f として, 次式を用いる.

$$f = a \times P - b \times L - c \quad (0 \leq a, b, c \leq 1)$$
- (4) エージェント優先度の要素である P_{ret}, P_{agt} が 1 つでも 1 以上になったら, モバイルエージェントは自分のエージェントサーバのもとへ戻る. そうでなければ, 次のサーバへ移動し (2) に戻る.

4. 動作実験

本論文で提案したエージェント優先度による行動制御方法を使用したモバイルエージェントと, 制御方法を使用しないモバイルエージェントの 2 種類を, 移動・分散プログラミング言語 Mobidget⁴⁾を用いて試作した.

4.1 実験環境

実験環境を以下に説明する.

- (1) 検索行動の結果, 新規検索データを保持することを模擬するためにエージェントサイズを 50 ずつ増加させる.
- (2) サーバ負荷度が小さい場合と大きい場合で検索行動時間を区別できるように, サーバ負荷度の増加にともなって検索行動時間は指数関数的に増加するものとし, 累積検索時間に加える.
- (3) 許容エージェントサイズは 400, 許容検索時間は 20000 ミリ秒と設定した.
- (4) 判断関数 f における a, b, c はそれぞれ 1.0, 1.0, 0.0 とした.
- (5) サーバ負荷度が大 ($L = 0.7 \sim 1.0$) のときの s, t, u は, それぞれ 0.5, 0.1, 0.4 とした.

表 2 移動先サーバリスト
Table 2 Server list.

サーバ名	サーバ負荷度
サーバ A	0.9
サーバ B	0.4
サーバ C	0.35
サーバ D	0.48
サーバ E	0.25
サーバ F	0.8
サーバ G	0.5
サーバ H	0.64
サーバ I	0.15

表 3 エージェント優先度を考慮しない場合
Table 3 Result without agent priority.

サーバ名	P	f	累積時間	データ量
サーバ A*	-	-	561	50
サーバ B*	-	-	15612	100
サーバ C*	-	-	19356	150

表 4 エージェント優先度を考慮した場合
Table 4 Result with agent priority.

サーバ名	P	f	累積時間	データ量
サーバ A	0.110275	-0.789725	411	0
サーバ B	0.347533	-0.0524667	852	0
サーバ C*	0.354867	0.00486667	1292	50
サーバ D	0.414296	-0.0657037	4580	50
サーバ E*	1	0.75	5051	100
サーバ F	0.368953	-0.431047	7647	100
サーバ G	0.478226	-0.0217741	8138	100
サーバ H	0.486576	-0.153424	8639	100
サーバ I*	1	0.85	9160	150
サーバ A	0.497767	-0.402233	11244	150
サーバ B*	0.544306	0.144306	11825	200
サーバ D*	0.612502	0.132502	15639	250

P : エージェント優先度 f : 判断関数

また, 実験用モバイルエージェントは移動先サーバリストに記述されている順番で巡回する. 各移動先のサーバにはサーバ負荷度が設定されている. 表 2 に, 実験に使用した移動先サーバのリストと負荷度を示す.

4.2 実験結果

4.2.1 エージェント優先度を使用しない場合

表 3 に, エージェント優先度を使用しない場合の実験結果の一例を示す. なお, 表中の * を付けたサーバは, モバイルエージェントが検索を行ったサーバであることを示す.

エージェント優先度を使用しない場合, モバイルエージェントは表 2 のリスト順で検索行動を行う. このため, サーバ A のサーバ負荷度が 0.9 (負荷大) であり, サーバ B に到達するまでに検索許容時間の大半を占めることとなり, その結果, 3 個のサーバしか巡

回できず、サーバCで検索を行っている間に検索許容時間である20000ミリ秒を超えてしまった。

4.2.2 エージェント優先度を使用した場合

次に、エージェント優先度を使用した場合の実験結果の一例を表4に示す。なお、表中の*を付けたサーバは、モバイルエージェントが検索を行ったサーバであることを示す。

エージェント優先度を使用した場合、モバイルエージェントはサーバに到着したときにその時点でのエージェント優先度とサーバ負荷度、判断関数を用いてそのサーバでの行動を決定する。表4に示すように、エージェント優先度を使用していない場合に比べて、より多くのサーバに巡回することができ、また検索行動もより多数のサーバで行うことが可能となっていることが得られた。しかし、サーバ負荷度が大きいサーバで検索せず、次のサーバに移動している。

5. おわりに

WWW検索において、時間制約下でモバイルエージェントが作業状況を考慮しながら移動・検索ができるように、モバイルエージェントにエージェント優先度を設定し、サーバ負荷度と組み合わせて行動制御を行う方法を提案した。これにより、モバイルエージェントの置かれている状況による制御ができるようになり、より多くのサーバで検索行動をとることができた。今回の実験では、サーバ負荷度を一定としたほか、検索結果のデータ量もサーバ負荷度によらず一定とした。今後、上記の点を動的に変化させた詳細なシミュレーション実験を行う予定である。また、この制御方法を

実際のWWW検索において使用できるように、巡回以外の移動に対する行動制御方法、利用者の要求を反映したWWWサーバごとの検索重要度の導入、企業内ネットワークでのエージェントシステム⁵⁾への応用拡大などをさらに検討する予定である。

参考文献

- 1) Kato, K., Someya, Y., Matsubara, K., Toumura, K. and Abe, H.: An Approach to Mobile Software Robots for the WWW, *IEEE Trans. Knowledge and Data Engineering*, Vol.11, No.4, pp.526-548 (1999).
- 2) 染谷祐一, 安部洋丈, 松原克弥, 東村邦彦, 加藤和彦: モバイルオブジェクトシステム PLANET を用いた Web 検索ロボットのモバイル化, 情報処理学会研究報告, 98-DBS-116, pp.25-30 (1998).
- 3) Papastavrou, S., Samaras, G. and Pitoura, E.: Mobile Agents for WWW Distributed Database Access, *Proc. 15th Intnt. Conf. on Data Engineering* (1999).
- 4) 藤田 悟, Jagannathan, S., Kelsey, R., 小山和也, Philbin, J., 山之内徹: 移動・分散プログラミング言語 Mobidiget: 言語仕様, 第57回情報処理学会全国大会論文集(分冊1)(1998).
- 5) Dasgupta, P., Narasimhan, N., Moser, E.L. and Melliar-Smith, P.M.: MAgNET: Mobile Agents for Networked Electronic Trading, *IEEE Trans. Knowledge and Data Engineering*, Vol.11, No.4, pp.509-525 (1999).

(平成12年6月5日受付)

(平成12年9月7日採録)