

モバイルエージェントの動きを利用した位置管理

小林 晋 輔[†] 石 井 雅 将[†]
小野里 好 邦[†] 河 西 憲 一[†]

モバイルエージェント (Mobile Agent: MA) はコンピュータネットワークに接続されたホスト間で自律的に移動することができるソフトウェアモジュールである。モバイルエージェントは各々がそれぞれの作業を実行するだけでなく、コンピュータネットワークを介した通信により、複数のモバイルエージェントが互いに協調して1つの作業を達成することも可能である。モバイルエージェント間の通信を実現させるためには、モバイルエージェントの位置情報、すなわちモバイルエージェントが存在するホストの情報を検索できる位置管理方式の開発が必要となる。本研究ではモバイルエージェントの振舞い、換言するとモバイルエージェントのホスト移動パターンを利用した位置管理方式を提案する。本提案方式は、モバイルエージェントが頻繁に訪問するホストを決定し、そのホストからモバイルエージェントの検索を開始することを特徴とする。提案方式ではモバイルエージェントの増加に対するスケーラビリティが確保され、かつ検索に要するコストの削減が見込まれることをシミュレーション実験により確認する。

Location Management of Mobile Agents Based on Their Behavior

SHINSUKE KOBAYASHI,[†] MASAYUKI ISHII,[†] YOSHIKUNI ONOZATO[†]
and KEN'ICHI KAWANISHI[†]

A Mobile Agent (MA) is a software module that is able to migrate autonomously from host to host in a computer network. Not only mobile agents can handle their own task individually, but also mobile agents can achieve a single task cooperatively by communicating with each other over a computer network. In order to realize communication between mobile agents, it is necessary to develop location management schemes that enable mobile agents to retrieve information about the location of mobile agents, i.e., the host in which mobile agents reside. In this paper, we propose a location management scheme that exploits behavior, or movement pattern between hosts, of mobile agents. A salient feature of the proposed scheme is that it determines the host that a mobile agent visits frequently, and begins to search the mobile agent from the host. By computer simulation, we confirm that the proposed scheme can reduce the cost of searching mobile agents while it preserves scalability against increasing number of mobile agents.

1. はじめに

モバイルエージェント (Mobile Agent: MA)¹⁾ は、ネットワーク中を自律して動くソフトウェアモジュールである。その動作は一度ユーザの手から離れるとその実行を終了するまでユーザのアクションや接続すらも必要とせず、自ら考え行動する。さらに状況の変化に応じて実行ホストの移動や、他のエージェントと連携することによって、ユーザから与えられたタスクを実行する。MA を実装するためのソフトウェアプラットフォームには、まだいくつかの課題が残されている

が⁴⁾、現在では e-コマースや情報検索をはじめとした分野で実用化され始め、大規模な MA システムの効率的な運用を可能とする技術開発が望まれている。そのためスケーラビリティの確保が重要となっている。MA 技術は、多くのネットワークアプリケーションに対し適合可能であり、なおかつ絶え間なく変動するネットワークの状況に柔軟に対応することが可能である。また、MA は容易に加工可能であり、ユーザごとに役割を特化させたエージェントを持つことができる。それにより、ユーザの代理人として様々な処理をしてくれるため、ユーザは複雑な作業から解放される。MA が複雑な処理を代行してくれるため、機能が限定された端末、たとえば携帯電話などからも高度な処理が実行可能となる。

[†] 群馬大学工学部情報工学科
Department of Computer Science, Faculty of Engineering, Gunma University

MA の能力の 1 つとしてコミュニケーション能力が存在する．そのため、他のエージェントとの連携や情報の共有などが可能となり、与えられた処理をスムーズに実行することができる．このことから MA システムはユーザ間で大きなリソースとデータベースを共有することになり、ネットワークトラフィックさえ考慮しなければタスクの実行に非常に有効な手段である．しかし、その自律性の高さゆえ、一度ユーザの手から離れるとその特性上位置を特定するのが困難なため、MA はコミュニケーション能力を活用することもできず、またユーザが想定する動作をしているか監視することも難しい．MA の位置登録実行時には、移動し続ける MA の探索は速やかに行う必要があり、同時に、MA の自由な移動を妨げないために、その位置登録は煩雑であってもならない．さらに多数の MA が動作するシステムを想定し、ボトルネックを生じやすい機構であってはならない．したがって、MA の位置管理を行うには、負荷が一点に集中せず、かつ検索をできる限り速やかに行える機構が必要となる．

そこで本研究では、MA の位置登録とその検索に MA の過去の振舞いを利用した移動ログベースホスト管理 (Migration Log managed by Base host, MLB と略す) 方式を提案する．ここでいう MA の振舞いとは、過去の動作から得られるホスト移動パターンであり、MA の持ち主であるユーザの志向や、MA のタイプにより異なる．MLB 方式では位置情報だけでなく、MA の振舞いを利用することにより、スケーラビリティを確保しながらも既存の方式より少ないコストで MA を見つけ出すことが可能であり、検索コスト面でも有効であることをシミュレーション実験で確認する．

以降、2 章で関連研究、3 章では MLB 方式における MA の位置管理方法、4 章において MLB 方式における位置検索コストのシミュレーション実験方法および評価、5 章で結論を述べる．

2. 関連研究

2.1 代表的な位置管理方式

MASIF³⁾ (Mobile Agent System Interoperability Facilities) は、OMG (Object Management Group) によって進められている MA システム間の運用に関する標準化案である．ここでは以下の 4 つの MA の位置検索方法が定義されている．

- ブルートフォース検索 (Brute force search)
- ロギング (Logging)
- MA の位置登録 (MA registration)

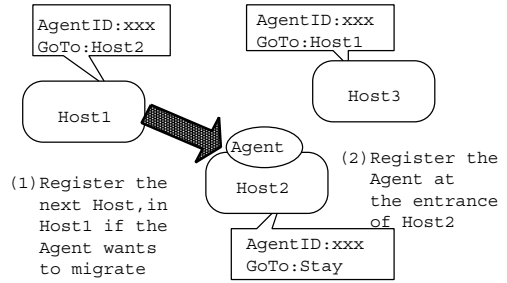


図 1 ロギングによる位置登録例

Fig. 1 An example of location registration by the Logging scheme.

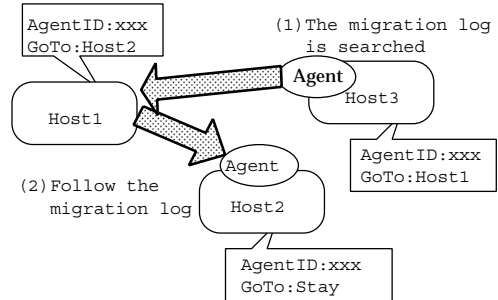


図 2 ロギングによる位置検索例

Fig. 2 An example of search procedure by the Logging scheme.

● MA の位置登録通知 (MA advertisement)
 ブルートフォース検索は、システム内すべての MA サーバに対して検索メッセージを送る方法である．当然のことながら、この方法では無駄となる検索メッセージも多く、システムの規模が大きくなればなるほど、ネットワーク全体にかかる負荷は増大する．

ロギングとは位置検索を行うときに移動ログを頼りに次々と移動先ホストを検索し MA を見出す方法をいう．ここで、移動ログ (Migration Log) とは MA が動作ホストで実行を終えてから別のホストへ移動する際に、動作ホスト内に登録した移動先のアドレスをいう．ロギングによる位置登録例を図 1 に、位置検索例を図 2 に示す．ロギングは位置登録に要する通信コストが低く移動の自由度は高いが、以下のような問題点があげられる．

- MA が移動を繰り返すと位置検索にかかるコストが増大する．
- 自分のいるホストに移動ログが発見できなかった場合ブルートフォース検索に頼らざるをえない．
- ホスト数が増えると検索が困難になる．
- ホスト内に蓄積していく移動ログの処理が問題になる．

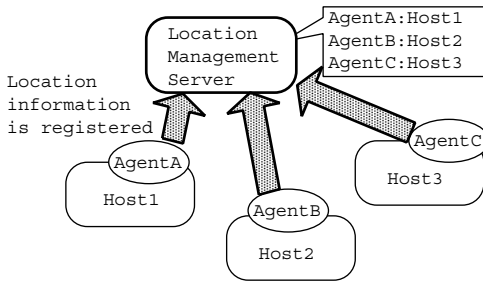


図 3 中央管理による位置登録例

Fig. 3 An example of location registration by centralized scheme.

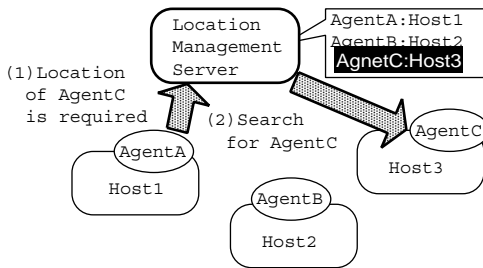


図 4 中央管理による位置検索例

Fig. 4 An example of search procedure by centralized scheme.

MA の位置登録では MA は移動のたびにその位置登録を特定のサーバへ行う。この方式の一例として中央管理方式がある。中央管理による位置登録例を図 3 に、位置検索例を図 4 に示す。中央管理は、相互通信においては高い性能を示すが、以下の問題点があげられる。

- MA の位置登録、検索ともに位置管理サーバ (Location Management Server) によって行われるため位置管理サーバの負荷が大きい。
- 位置管理サーバからのレスポンスが移動時間に影響する。

以上より MA の移動時にオーバーヘッドが増加することを考慮しなければならない。

MA の位置登録通知は、先の MA の位置登録亜種で、位置登録をしたい MA のみ、位置管理ホストへ位置情報を登録する。登録をしていない MA を探す場合にはブルートフォース検索を利用する。中央管理方式における管理サーバへの負荷は軽減されるが、MA のコミュニケーション能力の活用という視点からは、任意の MA との連携が困難となる。

上記の方式の中でロギング方式と中央管理方式の有効性の比較検討はマルコフ連鎖、シミュレーション、実環境において行われ、その結果によりロギング方式は MAN などのネットワーク、中央管理方式は LAN

などのネットワークにおいて有効であることが報告されている¹¹⁾。

2.2 dU-SSM

dU-SSM⁷⁾ は、一般的な位置管理方法である中央管理 (Centralized scheme) の位置情報更新頻度を移動ごとではなく、一定回数移動することによって実現される。dU とは d 回の MA の移動に対し、1 回のアップデート (Update) を行う方法である。そのため、位置情報検索と位置情報登録のオーバーヘッドを軽減し、大規模な MA システムで導入された場合でも、MA のモビリティが維持できる。SSM (Sequential Searching Method) は、更新されなかった移動を補完するもので、最後にアップデートされた場所から MA の探索を行うものである。

dU-SSM では、最新の位置情報更新から時間がたつにつれ検索コストが増大する。また、位置検索が頻繁に行われる MA に対しては、位置情報登録を頻繁に行わなければ、逆にオーバーヘッドが増大するという問題点もある。

2.3 SPC

SPC (Search-by-Path-Chase)¹⁰⁾ はネットワーク上にいる MA が存在する region ごとに ANS (Agent Name Server) によって管理する方式である。この方式は 1 つの region で見た場合は中央管理方式であるが、MA が別の region に移動した場合、現 region を管理する ANS は MA が移動した先の ANS を記録する。これにより region から移動先 region、移動先 region から MA が存在するホストという手順で MA を検索することが可能である。

この方式の特徴はモビリティはロギング方式、相互通信性は中央管理方式より優れてはいないが、モビリティと相互通信性の効率的なトレードオフにより平均的に高い性能を持っている点である。またスケーラビリティもある程度は確保している。

2.4 トラッキングエージェント

協調して作業を行う MA 間での互いの位置管理を対象としたものとしてトラッキングエージェント⁹⁾ と呼ばれる位置管理専用 MA を用いる方式がある。この方式では、協調作業を行う MA の位置情報をトラッキングエージェントが一括管理している。このトラッキングエージェントは協調作業を行う MA がマザーエージェントに生成要求を行うことで作成され、登録された各 MA の位置を管理する。マザーエージェントとは MA の要求に応じてトラッキングエージェントを生成する機能を有する MA である。トラッキングエージェントは管理する MA の位置情報を監視し、

MA の位置が物理的に偏った場合は自らの位置を移動させることにより MA 間の通信で発生する遅延を可能な限り抑える．トラッキングエージェントは登録されている MA の数が 1 以下になると自動的に消滅する．これにより位置登録にかかる通信コストを中央管理方式よりも少なく抑えることが可能である．トラッキングエージェントは登録されている MA の偏りに合わせて通信遅延が短くなる場所にいかにも移動するかが重要になってくるが、現段階ではその位置を決定する方法が確定されていない．そのため現段階では通信量が最も多い MA に近い場所へ移動するという方式が考えられる．

トラッキングエージェントは協調するエージェントどうしを管理するため、ネットワーク全体の MA を管理することは不向きである．また、協調する MA どうしがどのようにしてお互いを見つけるかについては研究段階である．しかし、発見後の協調作業時には優れた管理方式である．

2.5 IAgent

MA の位置管理に IAgent⁶⁾ (Information Agent) と呼ばれる位置管理専用の MA を用いた方法がある．この手法では、各 IAgent に複数の MA の位置情報を管理させ、管理する MA の数が増えれば IAgent 自身が分裂し、逆に減ったときには IAgent どうして統合することによって、動的に MA の位置管理を実現している．さらに、IAgent を管理するのに、ハッシュ関数を使うことによって、動的に IAgent の管理状態を変更可能としている．

この方法の最も大きな特徴は、位置管理に関する機構すべてを MA を使って実装することによる、スケーラビリティの高さである．しかし、IAgent に障害が起きた場合どのように位置情報を修復するか、また IAgent をシステムの状態によってどのように配置するかなどの課題も残されている．

3. MLB 方式における MA の位置管理方法

3.1 MLB 方式のアーキテクチャ

MLB 方式のアーキテクチャは以下の 3 つから構成される (図 5 参照)．

- ベースホスト (Base Host: BH)

BH とはネットワーク上で活動する MA の活動の中心となるホストである．MLB 方式による位置登録では、MA の振舞い (移動傾向) から BH となるホストを選び出す．BH となるホストは、MA が処理の実行を行うホスト (以下動作ホスト) のいずれかから選択され、その選択基準は、MA の

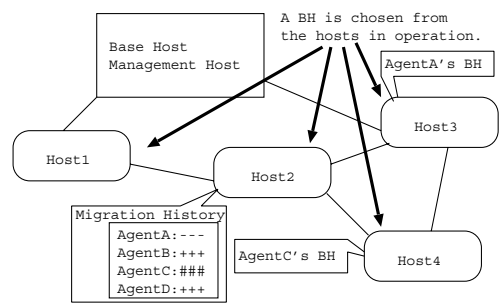


図 5 MLB 方式のアーキテクチャ
Fig. 5 Architecture of MLB scheme.

過去の実行において、訪れる頻度の高いホストとする．MA の動作に訪れる頻度の高いホストが見られない場合は、MA が生成されたホストを BH と設定する．また、仮に MA が生成され、処理を開始したホストと BH が異なった場合、まず MA は BH に現在位置を報告する．これは、移動をほとんどしていない MA のログを検索しやすくするためである．

- BH 管理ホスト (Base Host Management Host)

BH 管理ホストは BH がどこに存在するかという情報と、MA の ID やユーザの代行である MA の振舞いを保持するホストである．以降この情報を BH 情報とする．MLB 方式が、単純なロギングと異なる点は、各 MA が各々の BH を設定し、BH 管理ホストへの登録が必要となる点である．そのため各 MA の BH を管理するための BH 管理ホストが必要となる．
- 移動履歴 (Migration History)

移動履歴とは、現在動作ホストにおいて処理を実行中の MA が動作ホストに訪れたことを示す情報 (以降、入室ログ)、処理を終えて他のホストへ移動したことを示す情報 (退出ログ)、MA の移動先情報 (移動ログ) を保存したものである．この移動履歴を保持する機構を動作ホストに設定する．

3.2 MLB 方式における位置登録

図 6 は MLB 方式を用いた位置登録の例である．Host1 を BH とする AgentID:xxx が Host1 から Host2 へ移動する様子を示す．次の順で位置登録を行う．(1) AgentID:xxx は BH 管理ホストに BH 情報を登録する．(2) 他の動作ホストへ移動を行う場合はその行き先を移動ログとして残す．(3) 移動が完了したら移動先のホストに入室ログを登録する．

MLB 方式による位置登録の利点は、ネットワーク内の MA を管理するホストに対するアクセスを最小限

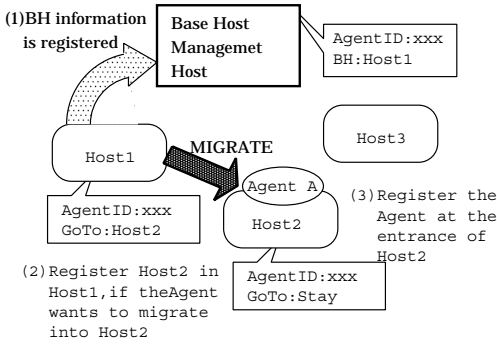


図 6 MLB方式に従って移動するエージェントの位置登録例
 Fig. 6 An example of location registration by MLB scheme when the MA whose BH is Host1 moves from Host1 to Host2.

に抑える点である。つまり、システム内の各 MA は、中央管理のように移動のたびに、BH 管理ホストに自らの位置情報の更新をする必要はない。BH 管理ホストにアクセスを必要とするのは BH の更新をするときのみである。位置情報の詳細は動作ホスト中に残すため、ネットワーク中に流れるパケットも中央管理に比べて少なくなる。

3.3 MLB方式における位置検索

位置検索は以下の順に行う。(1) BH 管理ホストに対してアクセスを行い、探したい MA の BH 情報を取得する。(2) この BH 情報をもとに MA の検索を行う。(3) BH にアクセスし、そこに MA が滞在していなかった場合、過去に検索対象の MA がその BH 上で処理を実行した履歴があるかどうかを調べ、もし存在したならば、その移動ログ情報をもとに次に検索を行う動作ホストを決定する。ただし、BH が複数個存在する場合、先に残りの BH を調べ、そこから得られた移動ログの中で最新のものをもとに検索する。

MLB方式では、BH への滞在がある程度期待できる BH を動作の基点にしているような MA において、その検索コストがロギング方式よりも低く抑えられる。図 7 に MLB方式による位置検索例を示す。図 7 は Agent B が Agent A を見つけ出すまでを表す。この例の場合、BH 管理ホストを訪れ、取得した BH 情報から BH を検索し、そこから 2 回動作ホストのログをたどることによって MA の発見に至っている。

3.4 MLB方式の改良

MLB方式では、BH での滞在率が低い場合、検索コストの問題は解決しないことが予想できる。そこで、MLB方式にアトリビュートを追加する。MA に一定回数の移動ごとに現在地の更新をさせることにより、検索のためにたどる移動ログが長くなりすぎるのを抑

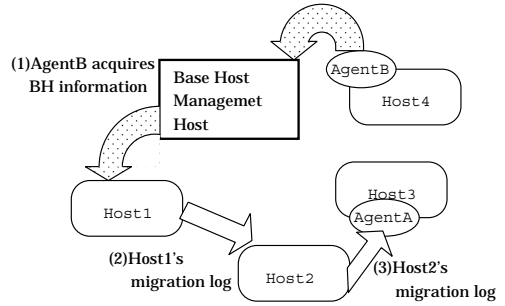


図 7 MLB方式による位置検索例
 Fig. 7 An example of search procedure by MLB scheme.

制する制限付き MLB方式 (Limited MLB: LMLB) を提案する。LMLB方式では、MA が BH から一定回数以上の移動を繰り返した場合、MA に現在地の報告を BH へ行うように設定する。MA が移動する際に許可する BH からの移動回数を表すアトリビュートを length として導入する。たとえば、length を 7 と設定した場合、BH から 7 回の移動まで BH へ現在地の報告をせずに移動することを許される。LMLB方式では位置情報の更新をさせるのが BH だけであるため、BH 管理ホストに負荷をかけることもない。また、BH 自体も各 MA ごとに決定されるため各ホストにかかる負荷も抑えることが可能である。

4. シミュレーションによる評価

4.1 シミュレーション実験方法

MLB および LMLB方式の有効性を確認するため、Aglets²⁾を用い、シミュレーション実験を行った。実験では処理の実行のためにネットワーク中を動いているエージェント A (図 6 参照) と、それを見つけ出すエージェント B (図 7 参照) を用意し、エージェント A の振舞いによって提案方式の検索コストがどのように変化するかを検証する。ここで検索コストとは、エージェント B がエージェント A の存在するホストに到達しエージェント A を発見するまでに、アクセスを必要としたホスト数を表す。比較のため、中央管理とロギングによる方法においても、シミュレーションを実行した。

本研究での位置登録コストは、位置登録の際にアクセスしたホスト数で考えられている。そのためロギング方式における位置登録コストは 1 移動につきコスト 1、中央管理方式も 1 移動につきコスト 1 を要する。MLB方式での位置登録コストは BH 登録とロギングのコストから構成される。BH 登録は BH 管理ホストへのアクセスだけで済むため、ロギング方式にコスト 1 を加えたものが MLB方式の位置登録コストとなる。

このため、既存の位置登録方式とのコスト差はつねに1であるため比較を行っていない。

MLB方式の場合、検索コストが最小となるのはエージェントがBHに滞在している場合である。そのときの検索コストは、BH情報を取得するためにBH管理ホストへのアクセスが1回、BHにエージェントが存在するかを確認するのにさらに1回、合計2回となり、以下移動ログをたどるごとに1ずつコストが増えていく。

同様に中央管理の場合の最小コストは2である。もしエージェントの位置情報を位置管理サーバから取得した後にエージェントAが移動した場合は検索失敗となる。このとき、もう一度位置管理サーバへ位置情報を取得するところから再開しなければならないため、検索が失敗するごとにコストが2ずつ増えていく。

ロギングの場合、最小コストはエージェントAがエージェントBのいるホストに滞在している場合でそのときの検索コストは1である。

今回、シミュレーションで用いるMAの移動パターンは、ランダムウォーク型(Random Walk Type)、巡回型(Round Type)、星型(Star Type)の3つである⁴⁾。それぞれの移動パターンにおいてBHは1つと仮定し、BHの変更は行わないものとする。図8に移動パターンの分類を図示し、表1にその仕様を表示した。

ランダムウォーク型では、ランダムにMAが動き、その移動パターンを分類できない。したがって、MAの振舞いからBHを設定することは困難であるため、そのBHはMAが生成され、動作を開始するホスト(Dispatched host)とする。ランダムウォーク型に分類されるアプリケーションの例として unstructured P2PにおけるMAを利用した検索方式があげられる⁵⁾。動的に変化するP2Pではランダムウォークによる検索も有効であることが示されている。

巡回型は、あらかじめ決まったホストを順に回っていく動作パターンを示すので、BHは巡回ホストの中から1つのホスト(Round host)を選んでBHとする。巡回型に分類されるアプリケーションの例としてeコマースなどがあげられる。eコマースなどはサービス側があらかじめ規定の場所に情報を提示しており、そのなかから情報を収集する形式であると考えられる。この場合、必要な情報を得ることができる経路を考え移動し、最終的にユーザのホストへ戻るため巡回型となる。

星型は、あるホストを基点として様々なホストへと移動をするパターンである。BHはそのような動作

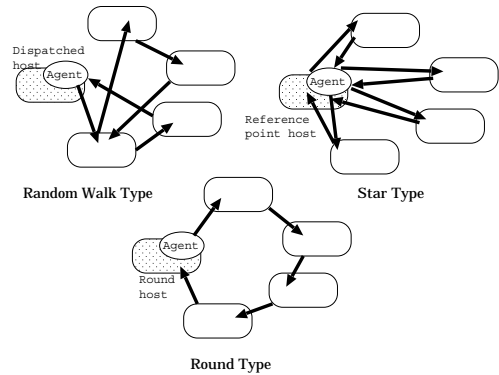


図8 MAの移動パターンの分類

Fig. 8 Classification of MA's migration pattern.

表1 移動パターンの分類

Table 1 Classification of MA's migration pattern.

	ランダムウォーク型	巡回型	星型
BH	生成ホスト	巡回ホストの1つ	基点ホスト
BH 滞在率	なし	各巡回ホストで一定	基点ホストと動作ホストの滞在時間の割合
移動先	ランダム	巡回先ホストから選択	基点ホストからの移動はランダム
想定アプリケーション	検索	eコマース巡回ナビゲーション	グループウェア家電の遠隔制御分散処理

の基点となっているホスト(Reference host)とする。その動作は基点ホストと他のホスト間の往復である。星型に分類されるアプリケーションの例としてグループウェアなどがあげられる。グループウェアで全体の作業を中央が割当てを行うような動作をするものや、並列して処理を行えないような作業ではリーダーと端末間での移動が行われるだけで端末間移動は行われぬ。そのため、その動作は星型となる。

4.2 シミュレーション実験の仮定

MA移動パターンは、先に述べた3種類とする。具体的には、ランダムウォーク型は各ホストへランダムに移動し、その滞在時間も平均 $1/\mu$ の指数乱数によって決定される。巡回型は、巡回先のホストをあらかじめ設定し、設定されたホストの中からランダムで移動をする。各ホストへの滞在時間はランダムウォーク型同様に決定され、その滞在率は巡回先のホスト数によって決定する。星型は基点ホストから動作を開始し、現在地が基点ホストなら次はランダムに移動先を決定する。巡回型と違い動作から滞在率を変動させることができないため、基点ホストとそれ以外のホストでの滞在時間の差によって滞在率が変動する。もし、現在

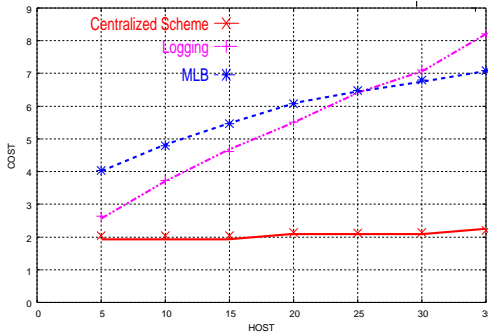


図 9 ランダムウォーク型における検索コスト対ホスト数特性
Fig. 9 Search cost versus number of hosts for Random Walk Type.

地が基点ホストでない場合、次の移動先ホストは必ず基点ホストとなる。

MA の動作するホスト数は 5 から 35 までとし、ホスト間の移動失敗は考慮しない。MA の各ホストでの滞在時間は、平均 $1/\mu$ の指数分布によって決定する。

MA の移動回数は、動作ホスト数に比例し、各ホストに平均 1 回以上移動することが可能な数にした。

実験データは、MA が処理を開始してから終了するまでのデータを取得した。

4.3 実験結果

4.3.1 ランダムウォーク型

図 9 はランダムウォーク型による各位置管理方法のホスト数対検索コストのグラフである。図 9 より、中央管理による検索コストはほとんど変動しないが、ロギングと MLB 方式はホスト数が増えるごとに検索コストが大きくなっていることが分かる。MLB 方式とロギングの 2 つを比較すると、ホスト数が少ない場合、ロギングの検索コストがより小さいが、ホスト数が増えるにつれその性能は逆転する。これはホスト数が少ない場合、ロギングで無作為にログを探していても移動ログが見つかる可能性が高く、MLB 方式において、BH 管理ホストから MA の BH 情報を得る利点がないからである。ホスト数が増えてくると、今度は無作為に検索することによってログを見つけるのが難しくなり、MLB 方式のコストが単純なロギングより良くなっていく。

時間ごとのコストの変動について図 10 を用いて考察する。図 10 はホスト数 25 のとき、単位時間あたり平均 1 回の移動がある MA のコストを示している。図 9 の長期的な検索コストの比較では、双方のコストはほぼ同じであったが、時間によるコストの変化を追っていくと違いが見えてとれる。図 10 より MA の実行間もない時間帯では、ロギングの検索コストは非常

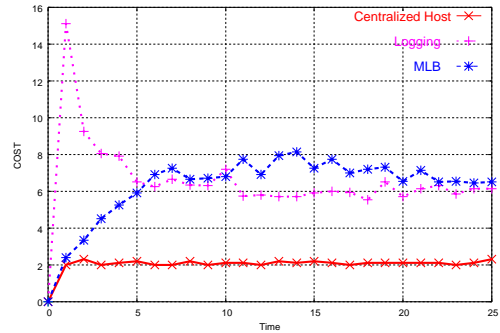


図 10 ランダムウォーク型における検索コストの時間依存特性 (ホスト数 25)

Fig. 10 Transient behavior of search cost for Random Walk Type when number of hosts is 25.

に高く、逆に MLB 方式は低い。そして、徐々に MLB 方式とロギングのコストは重なっていく。したがって、ランダムウォーク型 MA の検索の場合、MLB 方式は立ち上がりの検索、もしくは移動が少ない MA に適している。

4.3.2 巡回型

BH への滞在率を 10% とした場合のシミュレーション実験結果を図 11 に示す。ホスト数が増えても、MLB 方式では検索コストがほとんど変動していない。これにより、巡回型の振舞いをする MA に対して MLB 方式はある程度有効と考えられる。

図 12 はホスト数を 25 に固定し、巡回型 MA の BH 滞在率を変化させ、それぞれの滞在率における検索コストを示す。滞在率が増加すれば MLB 方式ではその分、検索コストが小さくなっていき、滞在率が 30% を超えるとほとんど中央管理と変わらない性能を示す。

時間の推移による影響を図 13 に示す。ランダムウォークのときと同様に MA が移動開始してすぐの段階での検索では、ロギングに比べて MLB 方式は優位である。

4.3.3 星型

図 14 は BH の滞在率 10% の場合における検索コスト対ホスト数のグラフを示す。ロギング方式はホスト数増加にともなって検索コストが増加しているが、MLB 方式の検索コストはホスト数に影響されず一定のコストを維持している。次に、BH 滞在率の影響を図 15 に示す。巡回型同様に滞在率が上がるにつれて、MLB 方式の検索コストは下がる。しかし、巡回型と違いその変化の幅は小さい。

以上より、星型の動作をする MA の検索において MLB 方式は高いパフォーマンスを示す。巡回型のときと比べても、さらに MLB 方式の検索コストは小さ

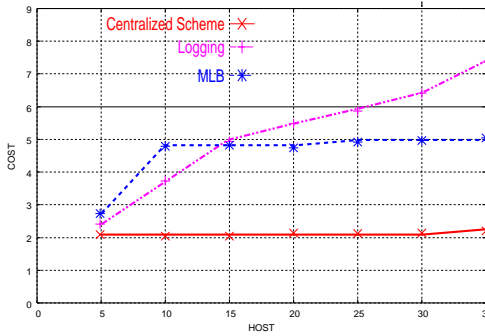


図 11 巡回型の検索コスト対：ホスト数特性（滞在率 10%）
 Fig. 11 Search cost versus number of hosts for Round Type (rate of stay 10%).

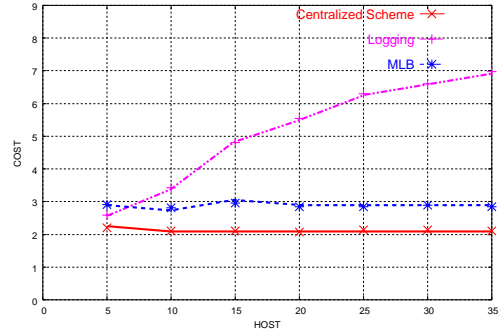


図 14 星型の検索コスト対ホスト数特性（滞在率 10%）
 Fig. 14 Search cost versus number of hosts for Star Type (rate of stay 10%).

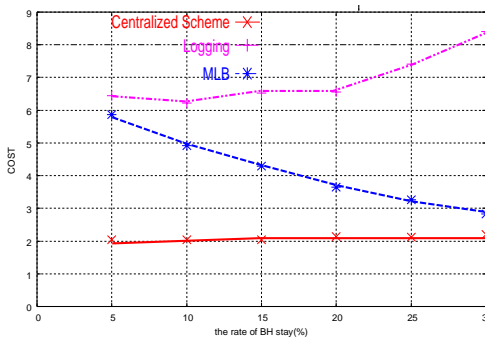


図 12 巡回型の検索コスト対 BH 滞在率特性（ホスト数 25）
 Fig. 12 Search cost versus rate of stay for Round Type (the number of host is 25).

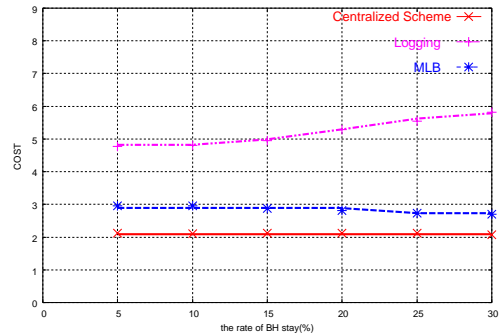


図 15 星型の検索コスト対 BH 滞在率特性（ホスト数 25）
 Fig. 15 Search cost versus rate of stay for Star Type (the number of host is 25).

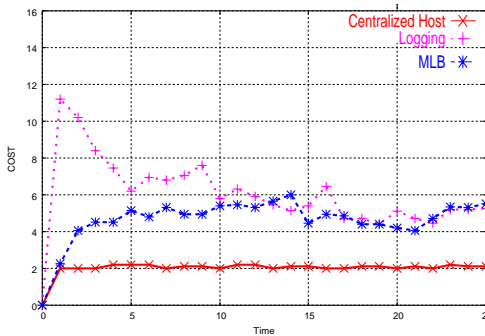


図 13 巡回型における検索コストの時間依存特性（ホスト数 25，滞在率 10%）
 Fig. 13 Transient behavior of search cost for Round Type (the number of host is 25, rate of stay 10%).

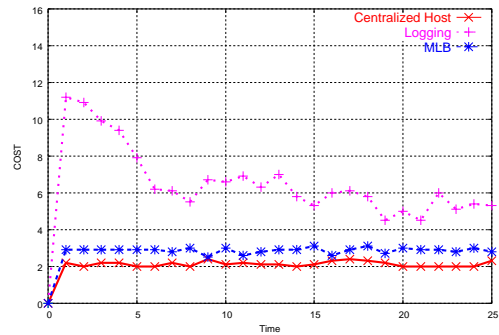


図 16 星型における検索コストの時間依存特性（ホスト数 25，滞在率 10%）
 Fig. 16 Transient behavior of search cost for Star Type (the number of host is 25, rate of stay 10%).

くなり、ほとんど中央管理による位置検索とコストが変わらない。

時間による影響を見ても（図 16）、MLB 方式は中央管理と変わらない検索コストとなっている。

4.3.4 LMLB 方式

図 17 と図 18 は LMLB 方式の実験結果である。

図 17 はランダムウォーク型の場合の検索コスト対時間特性を示す。移動制限なしの MLB 方式に比べ length 7 の制限でも十分にコストの削減が可能となる。さらに length の値を小さくしていくと、検索コストも下がっていくが、その下げ幅も徐々に小さくなっていく。

巡回型において LMLB 方式を実装した場合のグラフが図 18 である。図 18 より、length の値を小さく

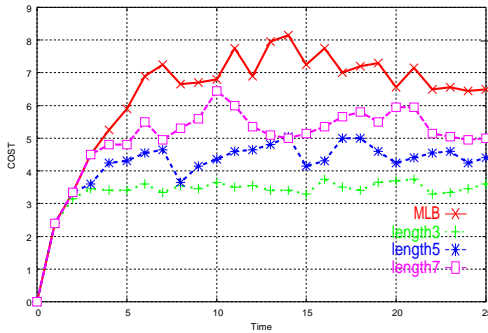


図 17 LMLB 方式のランダムウォーク型における検索コスト時間依存特性 (ホスト数 25)

Fig. 17 Transient behavior of search cost of LMLB for Random Walk Type (the number of host is 25).

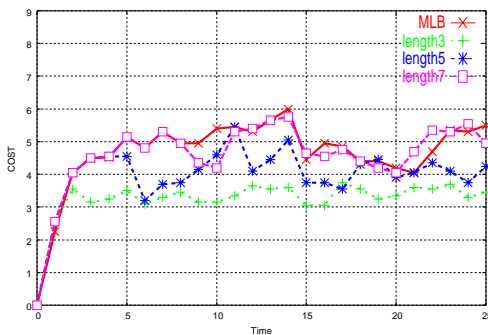


図 18 LMLB 方式の巡回型コストにおける検索コスト時間依存特性 (ホスト数 25, 滞在率 10%)

Fig. 18 Transient behavior of search cost of LMLB for Round Type (the number of host is 25, rate of stay 10%).

していけばいくほど検索コストの削減につながる。一方、ランダムウォーク型のときと異なる点は、length 7 のときの下げ幅が小さいことにある。これは、巡回型は同じホストを何度も回るためランダムウォーク型のときのように、移動ログが長くなり続けることが少ないからと考えられる。下げ幅は小さいが、ランダムウォーク型に比べて BH への報告回数も少ないため (表 2 参照)、巡回型において LMLB 方式を用いるときはある程度移動制限を厳しくとつても影響は少ない。また、巡回型については、BH 滞在率が高まるほど再報告の頻度が落ち、逆に滞在率が下がれば頻度は上がっていくため、移動パターンのみではなく、滞在率についても考慮する必要がある。また、LMLB では BH への再報告率に比例して位置登録コストが増大する。そのため、高い頻度で再報告が起きる場合には検索コストが下がったとしても必ずしも改善されているとは限らないことに注意すべきである。

表 2 ベースホストへの再報告の確率

Table 2 Probability of re-report to BH.

	ランダムウォーク型	巡回型 (10%)	巡回型 (20%)
length7	3.6%	0.4%	0%
length5	8.6%	2.4%	2.1%
length3	21.6%	9.6%	8.4%

4.4 考 察

シミュレーション実験より、MA の振舞いがパターン分けでき、かつある程度 BH への滞在率も確保できる場合、MLB 方式による MA の位置管理、およびそれを利用した位置検索は有効に働いていることが確認できた。

特に星型移動パターンの場合ほとんど中央管理と検索コストは変わらず、さらに中央管理で必要となる位置管理サーバの位置情報更新もほとんど必要がなくなるため、多数の MA から構成される大規模システムの場合 MLB 方式の方が有効である。また、MLB 方式は、巡回型においても、ある程度の BH 滞在率が確保できれば十分に有効な手段といえる。

逆にランダムウォーク型ではホスト数が増えるにつれ、巡回型では滞在率が下がるにつれ、MA の検索コストが増大する。これらの場合は、MA の移動制限を行うことにより、検索コストが軽減される。特に巡回型の場合ある程度制限をかけても BH への再報告確率は低い。したがって、頻繁に検索をかけられる MA の場合では、この移動制限を有効に利用するのが望ましい。

また、MLB 方式では、ロギングと同様に、各ホストに蓄積されていく移動ログ情報をどう処理していくかが問題になる。前述した移動制限を利用することにより、ホスト内の移動ログ情報にタイムアウトを設けることが可能になる。

MLB または LMLB 方式を導入する際にシステム内に付加されるオーバーヘッドとしては、MA の生成、削除時の BH 管理ホストへの登録、移動の際にホスト内に残す移動先、入室ログの登録あるいは、移動制限をかけられたときの BH への位置登録などがあげられる。これらの負荷はすべてシステム内に分散されるため、中央管理のように 1 点に負荷が集中することもない。また、ホスト数の増加にともなうコストの増加が少なく、高いスケーラビリティがある。

5. 結 論

本論文において MA の新しい位置情報管理方法を提案した。MA がユーザごと、役割ごとに多様な変化

をする点に注目し、提案方式では、その動作傾向を位置情報に組み込むことにより、検索コストの低減を試みた。シミュレーション実験では MA の移動パターンを大きく 3 つに分類し、それぞれについての検索コストを求めた。特に MA が特定のホストを多く訪れる場合において、MLB 方式は低いコストで検索可能であった。以上より、MLB 方式は、MA の動作に偏りがあればあるほど有効であり、機能ごと、個人ごとに様々な型に特化することが可能な MA において、MLB 方式を採用するのは有効である。

MLB および LMLB 方式が想定する環境としては、MA が星形の動作をするであろうソフトウェア、たとえば、MA によるチャットや、共同リソースを利用したソフトウェアの共同開発、あるいは情報検索用の MA などが考えられる。

今後の課題としては、より複雑な MA の動きに対して、提案方式は有効に働くかどうかの検証、また、複雑な動きをする MA の BH を、どのように決定するかなどがあげられる。また、提案方式では BH での滞在率が検索コストに大きく影響する。したがって、本論文では MA の生成時のみ BH 管理ホストへ登録をしたが、MA の動作状況に応じて逐次 BH の変更を可能とする基準の設定も必要である。

参 考 文 献

- 1) Puliafito, A., Tomarchio, O. and Vita, L.: MAP: Design and Implementation of a Mobile Agents Platform, *Journal of Systems Architecture: the EUROMICRO Journal*, Vol.46, No.2, pp.145-162 (2000).
- 2) Aglets. <http://www.trl.ibm.com/aglets/>
- 3) MASIF. <http://www.omg.org/>
- 4) 本位田真一: モバイルエージェントのためのソフトウェアプラットフォーム, *NII Journal*, No.3, pp.67-71 (Nov. 2001).
- 5) Gkanstsidis, C., Mihail, M. and Saberi, A.: Random Walks in Peer-to-Peer Networks, *IEEE INFOCOM 2004*, Hong Kong, March 7-11 (2004).
- 6) Kastidou, G., Pitoura, E. and Samaras, G.: A Scalable Mobile Agent Location Mechanism, *1st International Workshop on Mobile Distributed Computing (MDC'03)* held in conjunction with *23rd International Conference on Distributed Computing Systems (ICDCS'03)*, Rhode Island, USA, May 19 (2003).
- 7) Li, T.-Y. and Lam, K.-Y.: An Optimal Location Update and Searching Algorithm for Tracking Mobile Agent, *International Conference on Autonomous Agent*, Bologna, Italy, pp.639-646 (2002).
- 8) 小林晋輔, 石井雅将, 小野里好邦, 河西憲一: モバイルエージェントの動きを利用した位置登録, 電子情報通信学会技術研究報告, IN2003-145, pp.13-18 (Dec. 2003).
- 9) 國頭吾郎, 相澤清晴, 羽鳥光俊, 協調エージェント間通信のためのトラッキングエージェントの提案とその検討, 電子情報通信学会論文誌 (B), Vol.J81-B-I, No.11, pp.753-761 (1998).
- 10) Di Stefano, A. and Santoro, C.: Locating Mobile Agents in a Wide Distributed Environment, *IEEE Trans. Parallel and Distributed Systems*, Vol.13, No.8, pp.844-864 (Aug. 2002).
- 11) Alouf, S., Huet, F. and Nain, P.: Forwarders vs. Centralized server: An Evaluation of two Approaches for Locating Mobile Agents, *Performance Evaluation*, Vol.49, Issue 1-4, pp.299-319 (2002).

(平成 16 年 5 月 13 日受付)

(平成 16 年 11 月 1 日採録)



小林 晋輔

2002 年群馬大学工学部情報工学科卒業。2004 年群馬大学大学院工学研究科情報工学専攻終了。学生時代の研究テーマは「モバイルエージェントの位置管理」について。



石井 雅将

2003 年群馬大学工学部情報工学科卒業。現在、群馬大学大学院工学研究科情報工学専攻博士前期課程在籍。モバイルエージェントの位置管理について研究。



小野里好邦 (正会員)

東北大学卒業。工学博士。現在、群馬大学工学部情報工学科勤務。情報通信ネットワークに関する研究に従事。電子情報通信学会、日本シミュレーション学会、日本 OR 学会、IEEE、ACM 会員。



河西 憲一

1993年東北大学理学部物理学第二学科卒業．1995年東京大学大学院理学系研究科物理学専攻修士課程修了．同年日本電信電話株式会社入社．現在，群馬大学工学部情報工学科助手．博士（情報学）．情報ネットワークの性能評価に関する研究に従事．日本オペレーションズ・リサーチ学会，電子情報通信学会各会員．
