ﾏﾙﾁﾒﾃﾞｨｱ通信と分散処理 74－1
グループウェア 15－1
（1996. 1. 25）

移動型エージェントと静止型エージェントの特徴について

隆 朋也　渡辺 尚　太田 剛　水野 忠則

静岡大学

Tatsuya Suda

カリフォルニア大学アーバイン校

計算機ネットワークの世界的な拡大とユーザの増加によって、膨大な量の計算機資源や情報がネットワークによって接続され、広範囲に分散して存在している。しかし、ユーザがこれらを効率的に利用するためには、現時点ではユーザにネットワーク環境に関する多くの知識が要求されるのが実状である。この問題を解決するために、移動型エージェントと静止型エージェントを適宜使い分け、ユーザがネットワークに関する知識なしに適切なサービスが受けられる枠組みを提案する。本稿ではその準備段階として、それぞれのエージェントを定義し、それぞれの適応領域について考察する。また、エージェントの基本的な設計について述べる。

Characteristics of Mobile and Stationary Agents

Tomoya Taka　Takashi Watanabe　Tsuyoshi Ohta　Tadanori Mizuno

Shizuoka University

Tatsuya Suda

University of California, Irvine

Now so many computer networks are connected each other and spreading all over the world, and users of them are increasing, too. Therefore, we can access so many computer resources and informations through the computer networks. To use them efficiently, however, users are required many knowledges about the network environments. To solve this problem, we propose an agent system that allows users to receive appropriate services from computer network without any knowledges about that. In this paper, we work out two types of agents, mobile agents and stationary agents, and study about their adaptation field.

1　はじめに

近年の計算機ネットワーク技術の飛躍的な進歩や、異機種の計算機の接続を可能にするTCP/IPの普及によって、物理的に分散した計算機資源をネットワークを通して利用することが可能となった。現在では個々のネットワーク同士が相互に接続されることによって、計算機ネットワークは世界的な拡がりをみせている。また、計算機の高性能化、低価格化や、ネットワークに対するユーザの関心の高まりによって、計算機ネットワークを利用するユーザも増加の一途をた

どっている。ネットワークの拡大とユーザの増加によって、膨大な量の計算機資源や情報がネットワークによって接続され、広範囲に分散して存在している。

しかし、現時点では、ユーザが広範囲に拡がるネットワーク上のリソースを効率的に利用するためには、ユーザにネットワーク環境に関する多くの知識 (リソースの位置、特徴、利用状況、通信遅延の影響 etc.) が要求されるのが実状である。これらのすべてを把握することは困難なため、実際にユーザが有効に利用できるのは、ユーザが把握できている一部のリソースに限られてしまうのが現状である。

一方、システム側から見ると、ジョブの配送先の決定をユーザに任せることは、不要なトラフィックの発生や負荷の不均衡などの好ましくない現象を引き起こす原因となる。特に今後急速に増加することが予想される携帯型端末では、機能やメモリが限定された端末によって適切なサーバを選択せざるを得ないため、上記の問題がより発生しやすくなると考えられる。また、ユーザ層の拡大によって増加すると考えられるノービスユーザも、この問題を悪化させる原因となり得る。

このような問題を解決するために、エージェントを利用した通信方式が開発されつつある。これらのエージェントは、移動型エージェントと静止型エージェントの2つに大別することができる。前者はネットワーク中を移動しながらユーザの要求を達成するエージェントであり、後者はユーザの端末に存在して、メッセージを用いてサーバに仕事を要求するエージェントである。これらのエージェントを用いることによって、ユーザはエージェントに要求を渡すだけで、ネットワーク環境に関する付加的な知識を必要とせずに、結果を得られるようになる。

ただし、両者にはそれぞれ長所短所があり、得意とする適応領域が存在していると考えられる。ユーザが適切なサービスを受けるには、要求する処理に応じてエージェントを使い分けることが望ましい。

本研究は、この適応領域の差に着目し、移動型エージェントと静止型エージェントを適宜使い分け、ユーザがネットワーク環境を意識することなく適切なサービスが受けられる枠組みを開発することを目的とする。本稿ではその準備段階として、まず上記の移動型エージェントと静止型エージェントの定義を行い、次にそれぞれの適応領域について考察を行う。また、それぞれのエージェントの基本的な設計について述べる。

2 移動型および静止型エージェント

まず始めに、移動型エージェントと静止型エージェントの定義を行う。両者の最も大きな違いは、エージェント自身が移動するかどうか、である。したがって、ネットワーク上に距離、つまり通信遅延が存在しない場合には、この区別は意味を持たない。

移動型エージェント　移動型エージェントとは、ネットワーク中を移動しながらユーザの要求する処理を達成するエージェントを指す。つまり、要求を処理させるサーバまでエージェント自身が移動し、そこでサーバに処理を実行させるエージェントである。

移動型エージェントは、ユーザからの要求がシステムに投入されたときに生成され、サーバ間をメッセージとして移動する。エージェントは、ネットワークシステム内からユーザの要求を実行するのに最もふさわしいと思われるサーバを選択し、そのサーバまで移動してジョブを実行させる。そして、その実行結果をサーバから受け取ると、その結果に応じて次のサーバを選択し、そのサーバへと移動する。この動作を繰り返し、ユーザの要求が達成された時に、エージェントはユーザの端末へと帰り、得られた結果をユーザへと渡す。

静止型エージェント　静止型エージェントとは、ユーザの端末に存在し、ユーザの要求をメッセージによってサーバに伝え、その実行結果をユーザにフィードバックするエージェントを指す。つまり、自分は移動せずに、ネットワーク上の離れたサーバにメッセージによって仕事を依頼するエージェントである。

静止型エージェントはユーザの利用している端末に常駐し、ユーザからの要求を受け取る。エージェントはネットワークシステム内からユーザの要求を実行するのに最もふさわしいと思われるサーバを選択し、そのサーバへとジョブを依頼するジョブ要求メッセージを送信する。サー

表 1: それぞれのエージェントの特徴

	移動型エージェント	静止型エージェント
エージェントサイズ	小さい方が有利	大きくてもよい
利用するサーバ数	多くのサーバを使う場合	少ないサーバを使う場合
挙動の把握	困難	容易
ユーザとの通信	比較的困難	容易
ユーザ特性の学習	他の機構が必要	学習が可能
エラー処理	途中結果の取得が困難	途中結果の取得が可能
実装	全サーバがエージェントに対応する必要あり	従来の RPC, IPC などの技術を利用可能
ユーザ端末の負荷	ユーザ端末に負荷がかからない	ユーザ端末の負荷を上昇させる

バでの処理結果が返送されると、エージェントはその結果に応じて次のサーバを選択し、そのサーバにジョブ要求メッセージを送信する。エージェントはこの動作を繰り返し、最終的な結果をユーザへと渡す。

現在までに、この分野に関するいくつかの概念が提案されている。例えば、移動型エージェント記述言語 Telescript[1] や、メッセージに自律性とコード移動性を持たせた Messenger[2] などは移動型エージェントとして分類することができる。また、谷内らが提案した動的負荷分散システムで用いられるジョブスクリプト [3] も、移動型エージェントに分類できる。このシステムでは、ジョブスクリプトには実行すべきジョブがリストアップされ、スクリプト自体がサーバ間を移動しながら、それぞれのサーバでジョブを処理していく。一方、静止型エージェントとしては、ユーザの電子メール処理を学習し援助するエージェント Maxims[4] や Usenet Netnews のフィルタリングを援助するエージェント NewT[4] などが提案されている。ただし、これらはいずれも移動型もしくは静止型であり、その両者を組み合わせる発想はない。

3 エージェントの適応領域

移動型エージェントと静止型エージェントにはそれぞれ長所短所があり、その適応領域が存在していると考えられる。したがって、ユーザが要求する処理に応じてこれらのエージェントを使い分けるには、それぞれのエージェントの特徴と適応領域を把握することが重要になる。

それぞれのエージェントの特徴を表 1に示す。

3.1 移動型エージェントの特徴

移動型エージェントの利点として、ユーザ端末に負荷が集中しない点があげられる。ユーザ端末に常駐する静止型エージェントとは異なり、サーバへと移動して処理を実行する移動型エージェントは、ユーザ端末のリソースを占有しない。このため、携帯型端末のような機能やメモリが制限される端末でも、エージェントを利用してサービスを受けることが可能となる。

また、多くのサーバを利用する場合にも、エージェント自身がコンパクトなものであるなら、サーバとメッセージをやりとりするよりも、エージェントが移動する方が通信量の点から有利である。

一方、移動型エージェントの欠点としては、ユーザ側からのエージェントの挙動の把握が困難になる点があげられる。エージェントはその場の状況に応じてネットワーク内を自律的に移動していくため、ユーザからはエージェントの現在位置や状況が把握しにくくなる。その結果、ユーザとエージェントとの通信は困難になる。

また、移動型エージェントはエージェント自体が移動体であるため、エージェントがコンパクトである方がより望ましいと考えられる。こ

のため、エージェントの持てる機能が制限される。複雑なアルゴリズムや多量のユーザ依存データ、途中結果のログなどを持ち歩くことは、現実的には困難だと思われる。

これらの条件から、移動型エージェントに適した仕事の例として、エージェントによるショッピングを考える。ユーザは、購入を希望する商品の商品名のリストと予算の総額などの購入条件ををエージェントに伝える。エージェントはいくつかの店を回って、それぞれの店での商品の価格を調べ、商品と店の最適な組み合わせを計算し、ユーザの提示した条件が満たされる場合には、エージェントはそれらの商品をそれぞれの店で購入する。条件が満たされない場合には、エージェントはその結果をユーザに報告し、ユーザは条件を再考して新しいエージェントを派遣する。

この仕事では、エージェントは、価格の調査と商品の購入のために、複数の店を何度も移動することになる。ただし、エージェントはあらかじめユーザから与えられた条件にしたがって行動するだけで、エージェント自身が複雑な判断を行う必要はない。また、処理実行中にユーザとの通信を行う必要もない。よって、この仕事は、移動型エージェントに適していると思われる。

3.2 静止型エージェントの特徴

ユーザがネットワークを利用する場合には、以下の3つの局所性が生じると考えられる。すなわち、(1) リソース局所性：ユーザが実際に利用するリソースは限定されている。(2) ユーザ局所性：ユーザが投入するジョブは、各ユーザごとに異なる性質を持つことが多い。(3) 時間局所性：同一のユーザによって連続して投入されるジョブは、似た性質のジョブが多い。静止型エージェントは常にユーザ端末に存在し、その要求を受け取るため、投入される要求の傾向などから、ユーザに依存するこれらの特性について学習することも可能である。

静止型エージェントの最大の特徴は、エージェントが常にユーザの端末に存在していることである。このため移動型エージェントの場合とは対称的に、ユーザは常にエージェントの挙動を監視することができる。その結果、ユーザがエージェントに指示を出したり、割り込みをかけて制御することが可能になる。

さらに、サーバ側に対エージェント用の特別な機能を追加しなくても、RPC(Remote Procedure Call)、IPC(Inter Process Communication) などの既存の技術を用いることによって、サーバ－エージェント間でのジョブの受け渡しが可能になる点も、静止型エージェントの利点である。

ただし、エージェントが常駐することによって、ユーザ端末のリソースの一部がエージェントに占有され、ユーザ端末の負荷を上昇させてしまうと考えられる。また、サーバ－エージェント間の通信が頻繁に行われることによってユーザ端末周辺の通信量が多くなってしまうという欠点も予想される。

このような特徴を持つ静止型エージェントの例として、Maes らによる電子メールエージェント Maxims[4] があげられる。このエージェントはユーザに代わってメールに優先順位をつけ、メールの削除、転送、保存などを行う。エージェントは過去にユーザがメールに対して行った処理からユーザの特性を学習し、メールの到着などの新たな状況に対して、ユーザの動作を予測する。エージェントの予測に対してユーザが評価を与えることによって、エージェントは学習を重ね、確信度の高い予測が得られた場合には、それを実行に移す。このように、Maxims はユーザの特性を学習し、ユーザとの同期的処理によって学習結果の精度を高めることによって、ユーザにより良いサービスを提供する静止型エージェントである。

3.3 適応領域の切り分け

以上のように、移動型エージェントと静止型エージェントはお互いに相反する特徴を持つため、それぞれのエージェントに適した問題領域も異なっていると考えられる。したがって、両方のエージェントが利用できる環境では、ユーザが要求する処理の性質に応じて、それぞれのエージェントを使い分けることが望ましいと思われる。エージェントの適切な使い分けを行う場合、ユーザの要求がどちらのエージェントの適応領域に含まれているのかを判断するために、両エージェントの適応領域の境界を明確にする必要がある。

各エージェントの適応領域を切り分ける指標

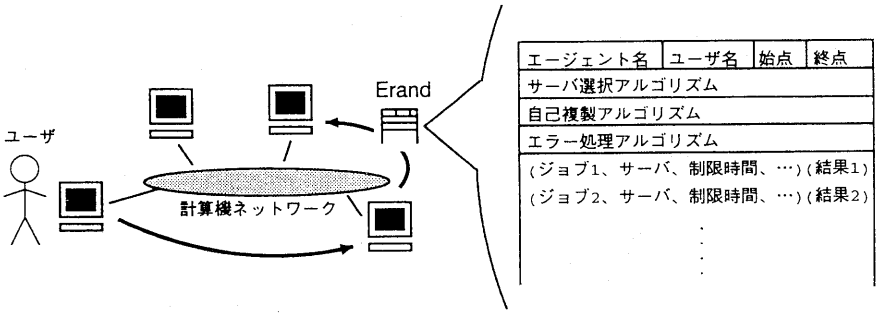

図 1: Erand の構成

として、以下のような項目を考える。ユーザからの要求をこれらの項目に関して分類することによって、その要求がどちらのエージェントに適した処理であるかを判断することができる。

- 利用するサーバ数と発生する通信量の関係
- ユーザとの同期処理の有無
- ユーザ特性の学習の必要性

移動型エージェントがサーバ間を移動するのに対して、静止型エージェントは要求メッセージとその返答メッセージを用いてサーバに処理を実行させる。したがって、移動型エージェントのサイズ、静止型エージェントが扱うメッセージのサイズ、通信遅延の大きさに応じて、移動型と静止型の適応領域の境界となるサーバ数が決定できると考えられる。

エージェントにユーザ特性を学習させる場合、エージェントが常に手元のユーザ端末に存在している静止型エージェントを用いる必要がある。また、ユーザがエージェントの挙動の監視、制御など行いたい場合にも、同様に静止型エージェントが適していると考えられる。これらの処理は、いずれも移動型エージェントで実現することはほぼ不可能なので、エージェントの選択時にはこれらの条件を利用サーバ数よりも優先する必要がある。

4 エージェントの基本設計

これまでに述べた移動型および静止型エージェントの特徴をもとに、それぞれのエージェントの基本的な構成を設計する。ここでは、自律的な移動型エージェント Erand と、ユーザ特性学習機能を持つ静止型エージェント Secretary を提案し、それぞれのエージェントについて、基本的な構成を設計する。

4.1 移動型エージェント Erand

移動型エージェント Erand はサーバ間で通信されるメッセージとして実装される。Erand は、ジョブシーケンス、サーバ選択アルゴリズム、自己複製アルゴリズム、エラー処理アルゴリズムから構成される。Erand の構成を図 1に示す。

ジョブシーケンスにはユーザの要求を達成するためのジョブがリストアップされている。ジョブは基本的にはこのリストに従ってシーケンシャルに実行される。ただし、前のジョブの結果に応じて処理を分岐したり、エージェントが自分自身の複製を作ることによって、複数のサーバで並列的にジョブを実行させることもできるものとする。

谷内らが提案するジョブスクリプト [3] では、次のジョブを実行するサーバの選択は、ジョブスクリプトが滞在しているサーバによって行われる。これに対して、Erand はサーバ選択アルゴリズムを持ち、前のジョブの結果に応じて自分で次のサーバを決定し、自律的にネットワークを移動する。

4.2 静止型エージェント Secretary

静止型エージェント Secretary はユーザの端末に常駐するプロセスとして実装される。Secretary は推論エンジン、環境観測機構、ユーザ依存情報ベース、HAI(Human-Agent Interface)、AAI(Agent-Agent Interface) の 5 つの構成要素

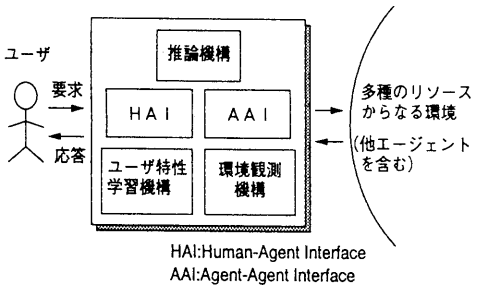

図 2: Secretary の構成

で構成される。Secretary の構成を図 2に示す。

推論エンジンは投入されたジョブに対して、最も適したリソースを選択するモジュールである。エージェントにジョブが投入された時に、推論エンジンは環境観測データとジョブ特性に基づいて推論を行ない、そのジョブの処理に最も適したネットワークリソースを選択する。推論に関しては推論時間、メモリ容量、CPU スピードなどの制限が生じることから、比較的推論が容易な Memory Based Reasoning(MBR)[5] を用いる。さらに、いくつかのケースを 1 つにまとめるルールを MBR に追加し、制約内での推論を可能とする。

環境観測機構はネットワークの環境情報を観測するモジュールである。観測した情報は内部のデータベースに格納され、必要に応じて取り出される。ここでいう環境情報とはネットワークで接続されたサーバの能力や利用状況、ネットワークの通信遅延の大きさ、他のエージェントの状態などを指す。

ユーザ特性学習機構には、ユーザが投入した要求について、その名前、性質、処理要求量、選択されたサーバ名などがコマンドレベルで記録される。推論エンジンはこれらの情報をもとに、ユーザの投入するジョブの性質の傾向や、それらの性質を持つジョブに適したリソース名を予測する。

HAI と AAI は、ユーザ－エージェント間およびエージェント－エージェント間のインタフェースを提供するモジュールである。前者には GUI などのマン・マシン・インタフェースなどが、後者にはエージェント間の通信プロトコルなどが含まれる。

5 おわりに

本稿では、効率的なネットワークサービスの利用のための移動型エージェントと静止型エージェントを定義し、それぞれのエージェントの適応領域に関する考察を行った。また、それぞれのエージェントの基本的な設計を行った。

本稿で提示したそれぞれの適応領域の切り分けについて、その正当性を検討することが今後の課題となる。そのために、まずエージェントの各構成要素、特に移動型エージェントのサーバ選択アルゴリズムと静止型エージェントの推論方式、環境観測と学習アルゴリズムを決定する。その上で、計算機シミュレーションや実装されたエージェントを用いて、さまざまな性質のジョブに対する各エージェントの性能を評価し、本稿で提示した適応領域の正当性について検討する。

また、より幅広いサービスを利用可能にするためには、ユーザの元まで移動してサービスを行う移動型サーバなど、サービスを提供する側のより多様なサービス形態についての検討も今後の課題として考えられる。

参考文献

[1] James E. White, "Telescript Technology: Mobile Agents," General Magic White Paper, 1995.

[2] Lubomir F. Bic, "MESSENGERS: Distributed Computing using Autonomous Objects," Technical Report 95-19, Dept. of Infomation & Computer Science, University of California, Ivine, 1995.

[3] 谷内, 太田, 渡辺, 水野: "分散システムにおけるジョブ特性を考慮したジョブの配送先決定法", 情報処理学会マルチメディアと分散処理ワークショップ, 1995.

[4] Pattie Maes, "Agents that Reduce Work and Information Overhead," Comm. of ACM, Vol. 37, No. 7, pp. 30–40, 1994.

[5] C. Stanfill and D. Waltz, "Toward Memory-Based Reasoning," Comm. of ACM, Vol. 29, No. 12, pp. 1213–1228, 1986.