

## 推薦論文

## 分散システム開発のための移動エージェントの定量的評価手法

前田直人<sup>†</sup> 中島 震<sup>†</sup>

分散システムの性能は、そのアーキテクチャに大きく依存する。アーキテクチャ・スタイルの選択には、性能に着目した定量的な評価に基づく方法を採用することが望ましい。本稿では、分散システム開発に移動エージェント技術を用いるか否かを判断するための1つの指針を与える定量的評価手法を提案する。従来の移動エージェントの定量的評価手法は有線ネットワークのみを想定し、無線ネットワークを考慮していなかった。しかし、近年、無線ネットワークを含む構成を持った分散システムは増加傾向にあり、評価を行う際、有線、無線ネットワークの性質の違いを考慮する必要がある。本稿では、Ethernet 10Base-Tと無線LANの2つの異なる性質を持ったネットワークを利用したワークフローシステムを例にとって我々の定量的評価手法を説明する。

## A Quantitative Evaluation Method of Mobile Agent for Distributed System Development

NAOTO MAEDA<sup>†</sup> and SHIN NAKAJIMA<sup>†</sup>

The performance of distributed systems greatly depends on their architecture. In order to choose a proper architecture style it is desirable to employ a quantitative evaluation method. This paper proposes a new quantitative evaluation method that clarifies a criterion whether we should use mobile agent technique or not. Previous work on performance evaluation for mobile agents has only assumed wired network environment, while wireless networks are getting popular. The difference between wired and wireless networks should be considered in the evaluation. This paper illustrates our quantitative evaluation method with a workflow application deployed in a heterogeneous network composed of an ethernet 10Base-T and a wireless network.

## 1. はじめに

分散システム開発時には、高い性能と拡張性の良さという相反する要求を満たすことが、設計のポイントとなる。特に、移動エージェント<sup>17)</sup>に代表される移動計算技術が、この問題を解決するソリューション技術として期待されている。移動計算は、最も高い実行効率が得られる「場所」に計算実体が移動することで性能を向上させる。同時に、移動させる処理内容を適宜変更することで、アプリケーション機能の拡張も可能とする。

移動計算技術は、移動する処理実体の使い方に応じて、Remote Evaluation (REV), Code on Demand

(COD), Mobile Agent (MA), などのアーキテクチャ・スタイル<sup>4)</sup>があり、適宜、選択する必要がある。特に、Aglets<sup>9)</sup>, MobileSpace<sup>14)</sup>, Plangent<sup>15)</sup>, MI-PLACE<sup>18)</sup>に代表されるMAは、うまく使いわけらることで、REVやCODの効果を実現することができ、移動計算の良いフレームワークを提供する。また、現実的なシステム構築にあたっては、移動計算技術と既存のクライアントサーバ型分散システムとの統合も重要な側面である。したがって、どのアーキテクチャ・スタイルをどこに用いるか、といったアーキテクチャ設計を分散システム開発の早い段階で行う必要がある<sup>12)</sup>。

アーキテクチャ・スタイルの選択には、利用する既存システムの使い方などのシステム要件に加えて、性能に着目した定量的な評価に基づく方法を採用するこ

<sup>†</sup> NEC ネットワーキング研究所  
NEC Networking Research Laboratories  
現在、法政大学経営学部  
Presently with Faculty of Business Administration,  
Hosei University

本論文の内容は2000年8月のオブジェクト指向2000シンポジウムにて報告され、SE研究会主催により情報処理学会論文誌への掲載が推薦された論文である。

とが望ましい。実際、移動計算技術を対象に、通信量を算定することで MA の使い方を議論した研究<sup>4)</sup>がある。しかし、アーキテクチャ・スタイルの性能評価に必要な通信時間の見積りに関する議論に欠けている。また、通信時間の測定により MA の優位性を議論する論文<sup>8)</sup>があるが、想定しているネットワークは有線であり、時間に依存して性能が変化する無線ネットワークを想定していない。近年、無線ネットワークを含む構成を持つシステムが増加傾向にあり、実際、MA の動機の一つは無線ネットワークが実現するユビキタス計算環境への対応であった<sup>17)</sup>。したがって、無線ネットワークでの定量的な評価を含めることが重要である。

我々が提案する定量的評価手法は、アプリケーションの性質や、配置する分散環境を考慮したシステム要件に基づいてエージェントの移動パターンを決定し、通信量を計算する。次に、利用するネットワーク性能を測定し、通信時間を計算することにより、ネットワークの特性や性能を考慮にいれた評価を行う。ネットワーク特性を考慮して移動エージェント (MA) の評価を行う点が本評価手法の特徴である。

本稿では、より具体的な議論を行うため、ワークフローシステム開発を例にとり、MA と CORBA (以下、RPC 型アーキテクチャ・スタイルと呼ぶ) の定量的評価を行う。

## 2. 移動の制御

MA の特徴は自律的な移動である<sup>17)</sup>。一方、システム開発の観点からは、MA の移動を制御することによりシステムの複雑さを軽減させることが重要な課題になっている。この章では MA の移動を制御する 2 つの手法を示す。

### 2.1 ビヘイビアパターン

MA は自律的な移動がその大きな特徴であるが、実行時に予測不能な振舞いをするにより分散システムの信頼性と効率に影響を及ぼすことが問題になっている。また実行時だけでなく、どのように移動するのかというエージェントの振舞いは分散システムの性能に影響を与えるため、振舞いが予測不能であると設計時にシステムの性能を見積ることが困難になる。その解決策として、ビヘイビアパターンが提案されている<sup>15)</sup>。ビヘイビアパターンはエージェントを利用したシステム開発の各段階で登場する再利用性の高いパターンを収集、分類、整理したものである。

本稿では文献 15) のビヘイビアパターンの特に移動パターンを利用する。4.4 節の通信量の評価では Itinerary と Star の 2 つの移動パターンを用いて評価

を行う。

### 2.2 Network Awareness

Network Awareness とはアプリケーションが動作しているプラットフォームのネットワーク資源の状態を知るといった性質を指す<sup>3)</sup>。ネットワークの状態を知り、状況に応じて MA の移動を制御することで、効率的で堅牢なアプリケーションが実現できる。近年、広く浸透しつつあるモバイル機器は、たびたび切断状態になり、性能が変動するような無線ネットワークを利用している。このような環境ではネットワークの状態を知ることが重要である。そこで、現在、移動エージェントシステムの中にネットワークやハードウェアの情報を提供する機能を追加し、環境を意識した MA を実現するための研究が行われている<sup>3),5)</sup>。

一方、Network Awareness を有効に活用するためには、異種混合のネットワーク上で定量的に移動エージェントの評価を行うことも重要である。本稿で提示する定量的評価方法のように、ネットワークの性質を調査し、MA の定量的評価を行うことで、ネットワークの状態に応じて、いつ、どのように振る舞えば良い性能が得られるのかといった指針を得ることができる。

## 3. ワークフローシステムの例題

提案する定量的評価手法を説明するために、例題としてワークフローシステム<sup>16)</sup>を利用する。ワークフローシステムは、その開発に MA を適用した例はあるが、MA の有効性が定量的に議論されていない。本章では、評価対象とするワークフローシステムの概要を述べ、次の章で評価を行う。

### 3.1 MA を利用する動機

MA の特性や利点に関しては文献 6), 10) に詳しい。ここでは特に、定性的側面からワークフローシステムの開発に MA を利用する動機について述べる。

#### 3.1.1 設計の容易さ

分散システムは複数の計算機にまたがって動作する。計算機の持つ資源や計算機を結ぶ通信路、物理的な配置などの様々な要因が分散システムの実装を制限する。問題領域をモデル化したときには 1 つの役割に 1 つのオブジェクトが対応していても、実装の段階では 1 つの役割が異なる計算機で動作する複数のオブジェクトに分割されてしまう。つまり、RPC を用いた設計では分割されたオブジェクト間でプロトコルを取り決めて通信を行う必要がある。一方、MA では移動する特別なオブジェクトの集合であるエージェントに 1 つの役割を割り当てることにより、設計と実装の意味的なギャップを小さくすることができる。

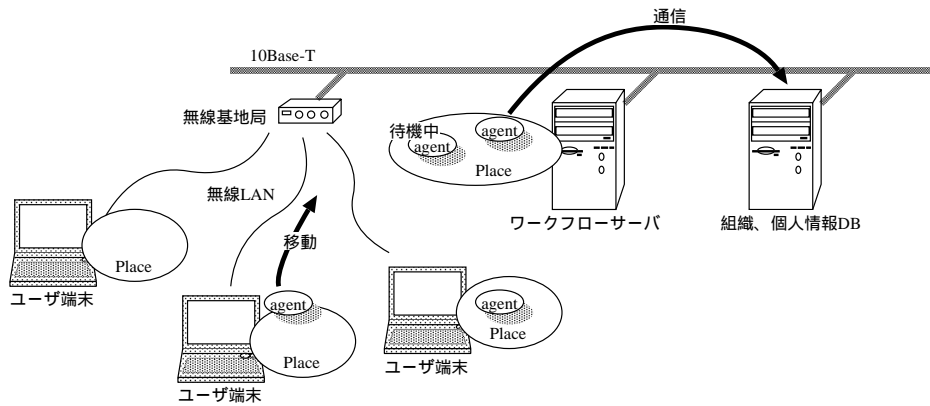


図 1 ワークフローシステム構成図

Fig. 1 Architecture of the workflow application.

### 3.1.2 コードの配信機能

多数のユーザが利用し、端末数が多い分散システムでは、アプリケーションのインストールやバージョン・アップに多くの手間がかかる。RPC の場合、クライアントとして動作するプログラムをすべての端末にインストール、設定しなければならない。一方、移動計算技術を用いることでコードの配信が可能になる。特に、MA は状態を持って移動するので、設定作業を自動化する効果も期待できる。ワークフローシステムの場合、MA を利用することで RPC と比較してクライアント端末の運用保守コストを低くおさえることが可能である。

### 3.2 ワークフローシステムの概要

本稿で対象とするエージェント技術を利用したワークフローシステムの構成を図 1 に示す。ワークフローシステムはワークフロー処理を行うワークフローエージェント、エージェントが動作するためのプレース、個人・組織情報の入ったデータベース、MASIF<sup>11)</sup> の実装コンポーネントから構成される。ユーザが利用する各端末には、エージェントが動作するためのプレースを配置し、サーバ側の計算機には、移動できなかったエージェントを受け入れるための待避用プレースや、ワークフロー処理に必要な組織情報、個人情報を格納したデータベースを配置する。ユーザが利用する端末は無線 LAN によってネットワークに接続しているノートパソコンであり、ネットワークから切断されている時間が長く、頻繁にシャットダウンされる性質を持っている。一方、サーバとして利用している計算機は 24 時間稼働しており、10Base-T の有線ネットワークに接続されている。ワークフローシステムの開発には Java<sup>1)</sup> を利用する。

MASIF はエージェントシステム間で相互運用する

ための標準仕様であり、エージェントシステム間で CORBA で接続するものである。MASIF はワークフローの重要な機能である進捗管理に必要なエージェントの制御や位置管理の機能を持っている。これにより MASIF のインタフェースを使ってエージェントの中断、再開、強制終了や、エージェント位置の発見が可能になる。また、MASIF の持つ管理ドメインの概念により組織の境界を明確にすることができる。このような理由から MASIF を採用する。

次に、簡単にワークフローシステムの仕組みを述べる。ワークフローサーバは待避中のエージェントを起動させる役割を持つ。そのためにユーザ・ログインの protocols を導入し、ユーザのログイン時に該当の待ち行列に入っているエージェントを起動する。同時に、ログイン時にユーザが利用している端末の場所を取得し、ユーザの位置管理を行う。さらに、ワークフローサーバはエージェント生成機能や進捗管理機能を持ち、クラスファイルや帳票用文書ファイルを集中管理する。ワークフローシステムのユーザはワークフロー管理用クライアントを使って、ログイン、帳票ワークフローエージェントの生成、生成したエージェントの進捗管理を行う。

### 3.3 帳票ワークフローエージェント

以下のような処理を行う帳票ワークフローエージェントを開発する：

- (1) 帳票ワークフローエージェントは購買要求を作成するユーザの端末に移動。
- (2) ユーザは帳票 (Word 文書) に記入。
- (3) 直属上司が利用する端末に移動。
- (4) 直属上司承認。
- (5) 価格が 100 万以上ならば、上位上司が利用する端末へ移動。

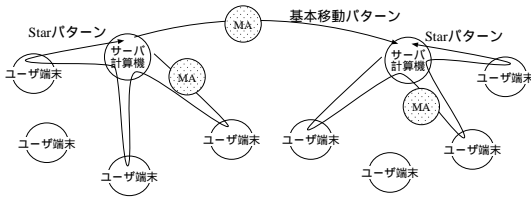


図 2 帳票ワークフローエージェントの移動パターン

Fig.2 Mobility pattern of the document workflow agent.

(6) 上位上司承認 .

(7) 購買課へ帳票を提出 .

ここで、エージェント移動のための論理的なトポロジを考える。先のシナリオから直感的に考えると移動パターンは上司や購買課を巡回する Itinerary になる。しかし、与えられた分散環境を考慮すると、帳票ワークフローエージェントの移動パターンは図 2 のような Star になる。理由は 2 つある。第 1 に、エージェントのユーザ端末への移動はつねに成功するとは限らない。そのため、移動に失敗したときにエージェントが待避する場所が必要になる。第 2 に、帳票ワークフローエージェントは移動先のユーザ端末上での処理中、もしくは移動中に障害により消滅する可能性がある。したがって、信頼性を高めるためには、移動する前にエージェントのバックアップをとっておき、必要に応じて複製を生成する必要がある。以上の理由から、移動パターンを Itinerary ではなく Star と考えた方がよい。待避場所となるサーバは特定の組織単位につき 1 つ存在し、各サーバ間を移動するために基本移動パターンも用いる。

4. 定量的評価方法

本章では我々の提案する定量的評価手法をワークフローシステム開発に対して適用し、実際に評価を行う。

4.1 評価手順

提案する定量的評価手法の評価手順を以下に示す：

- (1) アプリケーションの性質や、配置する分散環境を考慮したシステム要件に基づいてエージェントの移動パターンを決定 ( 図 2 を参照 ) .
- (2) 利用するネットワーク性能を測定 ( 4.2 節を参照 ) .
- (3) 通信量を評価 ( 4.4 節を参照 ) .
- (4) 通信時間を評価 ( 4.5 節を参照 ) .

アプリケーション設計者は、これらの評価を行うことにより、移動エージェント技術を採用すべきか否かを判断する 1 つの指針が得られる。設計者は、得られた指針と、与えられたシステム要件、アーキテクチャ・

表 1 無線 LAN の仕様  
Table 1 Specification of the wireless LAN.

規格	IEEE802.11, RCR STD-33
伝送速度	1Mbps/2Mbps
信号伝送方式	FH-SS
アクセス制御方式	CSMA/CA+ACK 方式

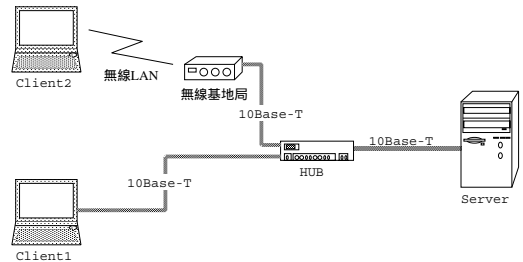


図 3 測定用の環境

Fig.3 Environment for the experiment.

スタイルの定性的特性を考慮し、最終的にアプリケーション開発に利用するアーキテクチャ・スタイルを決定する。

以降、ワークフローシステムの評価を通して、より具体的に我々の評価手法を説明する。

4.2 ネットワーク性能の測定

ワークフローシステムは、有線と無線の 2 種類のネットワークを構成要素として含む分散環境上に配置される。そこで、利用する各ネットワークの性能を測定する。図 1 に示したサーバ間を結び付ける有線ネットワークには Ethernet 10Base-T、ユーザ端末とサーバを接続する無線ネットワークには無線 LAN ( NEC C&C-NET Radio8300 ) を採用した。表 1 に無線 LAN の仕様を示す。

これらのネットワークの基本性能を測定するために、PC を 3 台用意し、1 台をサーバとして、他の 2 台はそれぞれ 10Base-T と無線 LAN を経由してサーバと通信を行った ( 図 3 ) . 各 PC の OS は Windows98、測定に利用した JVM は Sun JDK1.2 Classic VM ( build JDK-1.2-V, native thread, symcjit ) である。レイテンシ、スループットの測定はともに TCP を利用した。レイテンシは、1 byte のデータをクライアント / サーバ間で 10,000 回連続で往復させ、得られた時間を回数で割り平均を算出した。スループットは、100 byte から 1 Mbyte の間でデータサイズを変更し、1,000 回データを往復させた時間を測定して算出した。データサイズが 100 Kbyte のときに最大のスループットが得られた。また、Java を利用した場合の代表的な RPC 型のミドルウェアである RMI を利用して、1 回のメソッド呼び出しにかかる時間を計測した。Socket API

表 2 ネットワークの基本性能

Table 2 Basic performance of networks.

	レイテンシ	スループット	RMI
10Base-T	0.7 ms	1063.4 Kbyte/s	2.2 ms
無線 LAN	15.7 ms	80.5 Kbyte/s	17.7 ms

表 3 無線 LAN の測定結果

Table 3 Measurement on the wireless network.

転送量	通信時間 (最適)		通信時間 (不適)	
	平均	標準偏差	平均	標準偏差
6 Kbyte	167 ms	4.55 ms	3,080 ms	1,490 ms
10 Kbyte	266 ms	8.87 ms	3,100 ms	2,090 ms
50 Kbyte	1,230 ms	20.9 ms	5,200 ms	1,730 ms

を利用した場合に対する RMI 呼び出しのオーバーヘッドを知るため、レイテンシの測定と同様に、1 byte のバイト列を引数に与え、戻り値として同じく 1 byte のバイト列を返す遠隔メソッド呼び出しにかかる時間を測定した。RMI は 10,000 回連続して測定し、得られた時間を回数で割り平均を算出した。無線 LAN の測定は最も電波の受信状態が良い環境で行い、伝送速度として 2Mbps を選択した。表 2 に性能測定の結果を示す。

無線 LAN は電波の受信状態に応じて通信性能が変化する。そこで、さらに受信状態が最適な場合と最悪の場合に分けて、バイト列の往復転送時間を測定した。データを 50 回連続して往復させた時間を計測し、その時間を回数で割り平均を算出した。測定値のばらつきを示すため、同様の測定をそれぞれ 10 回行い、各平均値に対する標準偏差も算出した。表 3 に結果を示す。測定には先と同じ環境を利用し、Client2 と無線基地局の間に遮蔽物を置いて電波を遮る方法で接続不安定状態を作り出した。受信状態は無線 LAN に付属のネットワークモニタを用いて測定した。

#### 4.3 移動エージェントシステムの基本性能

アプリケーション開発のために、我々が実験用に試作した移動エージェントシステムを利用した。内部の実装をよく理解しており、他の移動エージェントシステムと比較してシンプルな構成であるため、性能の予測が容易であることが、その理由である。試作版移動エージェントシステムはエージェントを記述する枠組みとして Aglets<sup>9)</sup> と同様な Call-back 型イベントモデルを採用し、エージェント間通信用に非同期メッセージ機能を提供する。また、他の多くの移動エージェントシステム同様、独自にクラスローダを定義しており、エージェントの移動ごとにプログラムコードの転送が発生する。

エージェントを転送する処理の流れを図 4 に示す。

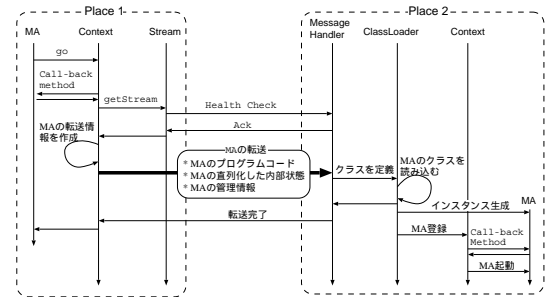


図 4 MA 転送シーケンス図

Fig. 4 Sequence of transferring MA.

表 4 エージェントの往復時間

Table 4 Agent roundtrip time.

	10Base-T	無線 LAN
試作版エージェントシステム	62.8 ms	167.8 ms

転送に利用する通信路が利用可能かどうかを確認する通信 (Health Check) を除けば、エージェントの転送は 1 回のリクエスト/レスポンスで処理される。転送するデータは以下の 3 つの要素から構成される。

- プログラムコード
- エージェントの内部状態
- エージェントの管理情報

プログラムコードはエージェントの実行に必要な Java クラスファイルである。エージェントの内部状態は Java の直列化 API を利用して直列化したエージェントオブジェクトである。エージェントの管理情報とは、エージェントの識別子、動作中/停止中などのエージェントの状態を示す整数、エージェント間通信用のキューを含んだ管理用オブジェクトを直列化したものである。

参考として、移動のみを行う最小のエージェントの往復時間を表 4 に示す。図 3 に示した環境を利用し、エージェントがクライアントとサーバ間を 1,000 回連続して往復するのにかかる時間を計測し、1 回の往復時間の平均を算出した。測定に利用したエージェントの転送データ量は、上記 3 つの要素をすべて含め 1781 byte である。

#### 4.4 通信量の評価

RPC と MA を用いてワークフロー処理を行った場合のそれぞれの通信量の評価を行う。特に MA は移動パターンによって通信量が増えるため、先に議論した Itinerary パターンと Star パターン<sup>15)</sup> について考える。

通常、ワークフローの処理には複数の人間が関わる。その中でワークフローを開始した人間をクライアント

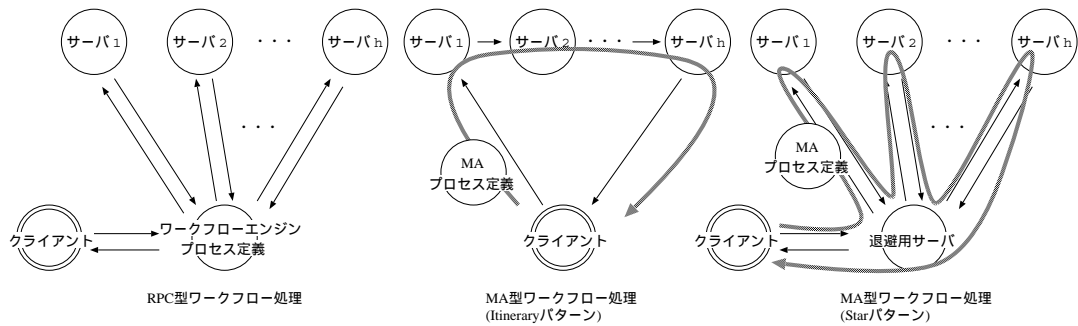


図5 RPCとMAのワークフロー処理モデル

Fig. 5 Models of RPC and MA workflow processing.

として、残りの人間をサーバと見なす。つまり、ワークフロー処理が人間を相手にしているのか、それともデータベースのような計算コンポーネントを相手にしているのかを区別しない。ワークフロー処理に必要なサーバはネットワークで接続された複数の計算機上で動作しており、その数を  $h$  とする。RPC 型のワークフローは特定の計算機上で動作するワークフローエンジンがワークフロー定義を解釈し必要なサーバと通信を行う。一方、MA のワークフローはエージェントがワークフロー定義とワークフローエンジンを内部に保持し必要なサーバに移動して処理を行う。図5にそれぞれの処理の様子を示す。クライアントが文書ファイルの巡回の開始点であり、終点である。簡単のため処理開始時に文書ファイルや MA のクラスファイルはクライアントの端末にあるものとする。

ワークフロー処理で扱う文書ファイルのサイズを  $d$  とする。RPC は各ホストとの間で遠隔メソッド呼び出しが発生するので、1 つのホストに対する呼び出し回数の平均を  $n$  とする。RPC の 1 回のメソッド呼び出しで発生する、引数および戻り値の転送量の和の平均を  $rpc$  とする。

一方、MA は移動のときのみ通信が発生する。4.3 節で示したように、我々の移動エージェントシステムがエージェントを移動させるときに送信するデータは、エージェントが動作するために必要なクラスファイルと、直列化したエージェントオブジェクト、直列化したエージェント管理オブジェクトを含む。この転送データ構成は特有ではなく、多くの移動エージェントシステムに共通のものである。これより、MA の移動で発生する転送量は、クラスファイルと、直列化したエージェントオブジェクトと、直列化したエージェント管理オブジェクトのデータ量の合計である。これを  $agent$  とする。通信量を算出する式は以下のようになる：

$$RPC = (d * 2 + n * rpc) * (h + 1)$$

$$MA(Itinerary) = (d + agent) * (h + 1)$$

$$MA(Star) = (d * 2 + agent * 2) * (h + 1)$$

一般に、MA は通信回数を削減する効果がある。RPC は各ホストごとに  $n$  回のメソッド呼び出しが発生する。コードの配信ができない RPC を利用してワークフローシステムの拡張性を上げるためには、汎用的な小粒度のサービスを多数持ったサーバをユーザ端末に配置する必要がある。つまり、拡張性を考慮すると  $n$  の値が増加する<sup>8)</sup>。一方、MA が必要とする通信はホストごとに MA(Itinerary) では 1 回、MA(Star) では往復なので 2 回だけである。

次に、前記式を用いて通信量の観点から評価を行う。まず、 $d > agent$  が成立すれば、MA(Itinerary) が最も通信量が少ないといえる。今回、帳票データ形式として、企業で実際によく利用されている Microsoft 社の Word 文書を採用した。Microsoft 社の Word 文書を利用した帳票をいくつか調べたところ、非常に簡単なもので約 20 Kbyte、複雑なもので 60 Kbyte 程度であった。一方、我々のエージェントシステムを利用した場合、巡回型エージェントの実行コードサイズは、経験的に、大体 10 Kbyte 以内に収まると予想された。これより、MA(Itinerary) が通信量の観点で最も性能が良いと判断できる。

一方、システム開発にあたり障害への対応を考えると、移動エージェント型のワークフローシステムでは MA(Itinerary) ではなく MA(Star) を採用する必要がある。そこで、次に MA(Star) と RPC の比較を行う。双方の文書データの転送回数は同一であるため、これらの比較評価は文書データサイズに依存しない。前記式より、MA(Star) が通信量の点で RPC より優位にたつための条件は  $n * rpc > agent$  であることが分かる。この条件を満たさない場合、 $(agent - n * rpc)(h + 1)$  が RPC に対する MA(Star) の通信量のオーバーヘッド

表 5 試作ワークフローシステムのデータ量  
Table 5 Data size of the workflow application.

	サイズ
文書ファイル	48.6 Kbyte
クラス	6.9 Kbyte
直列化データ	0.5 Kbyte

になる。我々は *agent* の値を 10 Kbyte 未満と見積った。したがって、 $n * rpc > 20$  Kbyte が通信量の観点から 2 つのアーキテクチャ・スタイルを使い分ける 1 つの目安になる。

我々は MA(Star) を採用したワークフローシステムを実際に試作した。そこで、参考として、試作した帳票ワークフローエージェントの文書ファイル、クラスファイル、直列化データのデータサイズを表 5 に示す。この値を利用すると、 $n * rpc > 14.8$  Kbyte のときに、MA(Star) は RPC より通信量の観点で性能が良いといえる。

#### 4.5 通信時間の評価

この節では、ネットワークの性質を考慮した通信時間の観点から MA(Star) と RPC の評価を行い、通信量の評価だけでは、性能を評価するために十分ではないことを示す。

システムが配置される分散環境、特にネットワークの性能は通信時間に影響を及ぼす。そこで、あるネットワーク環境  $E$  において、通信量  $c$  のデータを転送するためにかかる時間を返す関数  $f^E(c)$  を導入する。 $f^E(c)$  の計算には、実際にネットワークの性能を測定した値を利用する。

通信時間は、転送量に依存して変化するため、毎回の通信を区別して考える必要がある。したがって、RPC の通信量を見積る式では  $rpc$  は 1 回のメソッド呼び出しで発生するリクエストとレスポンスの転送量の和であったが、通信時間を評価する場合は、リクエストとレスポンスをそれぞれ分けて考えなければならない。そこで、 $rpc$  を各メソッド呼び出しで発生するリクエストおよびレスポンスの転送量の平均の値とする。 $d, n, h, agent$  は転送量の評価の式で用いた変数と同じものを示す。RPC は最初と最後にそれぞれ文書を転送することに注意する。

MA の通信時間を見積るためには、移動エージェントシステムの転送プロトコルを知る必要がある。4.3 節で示したように、我々の移動エージェントシステムは、1 回のリクエスト/レスポンスでエージェントを転送する。Health Check のために発生する通信は省略して考える。文書はエージェントの内部状態に含まれるのでエージェントと同時に転送される。

表 6 転送量あたりの転送時間  
Table 6 Communication delay taken to transfer the data.

転送量	1 byte	100 byte	40 Kbyte	50 Kbyte
転送時間	7.8 ms	8.8 ms	513 ms	625 ms

通信時間を見積る式を以下に示す：

$$RPC = (f^E(d) * 2 + f^E(rpc) * 2(n - 2)) * (h + 1)$$

$$MA(Star) = (f^E(d + agent) * 2) * (h + 1)$$

我々は以下のように各パラメータの値を見積った：

- $d$  : 40 Kbyte
- $agent$  : 10 Kbyte
- $rpc$  : 1 byte, 100 byte

$rpc$  は 1 byte と 100 byte の 2 種類の場合を考えた。

次に、無線 LAN 環境での通信時間を算出するために、各データサイズの転送に要する転送時間をそれぞれ測定した。測定結果を表 6 に示す。図 3 に示す測定環境で、TCP を用いて Client2 と Server 間のバイト列の往復時間を測定した。さらに、一方向の転送時間を計算するために、往復時間を 2 で割り、この値を転送量に対する転送時間とした。測定は 1,000 回行い平均値を算出した。転送時間は  $f^E(c)$  で利用し、 $E$  は無線 LAN とする。

通信量の計算式と、データサイズ、および転送時間より、RPC に対する MA(Star) の通信時間のオーバーヘッドを表す式  $(224 - f^E(rpc) * 2(n - 2))(h + 1)$  が得られる。 $rpc$  を 100 byte と見積った場合、RPC の 1 つのホストに対する呼び出し回数の平均  $n$  が 14.7 より大きければ、MA(Star) の通信時間の方が少なくなることが分かる。 $rpc$  を 1 byte とした場合、 $n$  は 16.4 となる。通信 1 回で発生する転送量が数十から数百バイトであるような RPC は、転送量の差より転送回数の差の方が性能に影響を与える重要な要因であることが分かる。

先の通信時間の評価で用いたデータサイズと 4.4 節の通信量の式を用いて、同様の比較を行うと、次の結果が得られる。 $rpc$  を 200 byte とすると  $n$  の値は 100 になる。また、 $rpc$  を 2 byte とすると  $n$  の値は 10,000 になる。先の通信時間からの評価から得た  $n$  の値と大きく異なる。

したがって、アーキテクチャ・スタイルの性能は、通信量ではなく通信時間によって評価されるべきである。従来の通信量の評価だけでは不十分であり、通信量に対するネットワークの通信性能を計測し、通信時間の評価まで行うことが重要である。

#### 4.6 考 察

先の評価により得られた指標をもとに、無線ネットワークを利用したワークフローシステム開発に MA(Star) を採用することが妥当であるかどうか考察を行う。

まず、通信量の評価より、MA(Star) が RPC と比較して優位になる境界を示す  $n * rpc > 20 \text{ Kbyte}$  という式が得られた。1 回のメソッド呼び出しで発生する転送量  $rpc$  の値を約 200 byte と見積ると、各ホストで発生するメソッド呼び出しの平均回数は 100 回以上である必要がある。通信量の観点からは、RPC の方が効率的であると判断するのが妥当である。

一方、通信時間の評価では、MA(Star) が RPC と比較して優位になる境界を示す  $n$  の値は 14 から 17 程度であるという結果が得られた。 $n$  の値は 14 より小さいと考える方が一般的であり、RPC の方が実行性能がよいと考えられる。しかし、通信量からの評価と比較して、使い分けの指針となる  $n$  の値は大幅に減少している。RPC を利用した場合の妥当な  $n$  の値を定めることは困難であるが、 $rpc$  を 100 byte、 $n$  が 3 から 10 程度であると仮定すると、MA(Star) は RPC と比較して通信時間が 1 割から 2 割増加すると予想される。

さらに、ネットワークの性能測定より得られた結果について考える。表 3 より、無線ネットワークでは次の 2 つの性質について考慮しなければいけない。まず、受信状態に応じて通信性能が変化するため、性能が一定ではない。時間によって通信性能が変化する性質を持つ。次に、受信状態が最悪の環境における測定値のばらつきが非常に大きい。数 Kbyte 程度の転送量の差を比較した場合、それぞれの転送量の平均の偏差より同じ転送量で測定した値の偏差の方が大きくなる。つまり、無線 LAN の電波の受信状態が悪い場合、そもそも転送量に対する通信時間からの議論が成立しない。

このような不安定な無線ネットワーク環境では、MA の他の特徴も考慮した方がよい。たとえば、帳票ワークフローで利用した文書ファイルの転送にかかる時間は、受信状態に応じて、約 1 秒から 8 秒のばらつきを見せる。この変化に対応できる機能を持つかどうかは性能に大きく関わる。さらに、無線ネットワークでは切断状態への対応も考慮しなければならない。MA の実装は、その自己完結性や自律性により、Network Awareness を利用した環境の変化への対応や、切断時の処理の続行を可能にするよいフレームワークを提供する。

以上より、MA(Star) は RPC と比較して通信時間が 1 割から 2 割程度増加すると予想されるが、時刻により性能が変動する無線ネットワークの性質を考慮した場合、無線ネットワークがもたらす諸問題を解決するよい枠組みを提供することのできる MA(Star) をワークフローシステムの開発に利用することは、有効であるとする。

#### 5. 関連研究

文献 4) では RPC、REV、COD、MA をモデル化して通信量に対する評価を行う手法の提案を行っている。例として情報検索システムの評価を行い、REV が RPC と比較して通信量が少ないことを示している。文献 8) では同様に、Internet を経由した通信が発生する 2 つの分散システムを用いて、通信時間の観点から RPC と MA の効果測定を行っている。クライアントとサーバ間での通信回数  $n$  や圧縮対象の文書ファイルのサイズが大きくなると RPC より MA の処理時間の方が短くなることが示されている。しかし、これらの研究では通信回数や転送データ量の増減が大きいアプリケーションのみを対象としている。また、これらの研究ではネットワークの性能や性質まで踏み込んだ議論をしていない。本稿では、ネットワーク特性を考慮した定量的評価手法を提案し、情報検索と比較してより一般的であると考えられるワークフローシステムに対して本手法を適用して、移動エージェント技術の有効性を議論している点で新しい。

文献 2) や文献 7) では、確率統計モデルを利用してアーキテクチャ・スタイルの評価を行っている。文献 2) では、従来のイベント配信アルゴリズムと、「広告」というアイデアを導入したイベント配信アルゴリズムを評価し、後者のアルゴリズムが優れていることを示している。イベント配信の頻度や経路はアプリケーションに依存するため、確率統計モデルを用いて大まかなシステムの振舞いを表現するモデルを作成し、評価を行っている。また、文献 7) はアーキテクチャ・スタイルの性能評価に待ち行列モデルを利用することにより、システムの平均応答時間やスループットの予測を可能にしている。一方、本稿の定量的評価手法では確率統計モデルを利用しない。本手法の目的は、一般的なアプリケーションに対する MA の有効性を示すことではない。個々の分散システム開発時に、システム設計者に、MA を利用することが妥当であるかどうかを判断する指針を与えることが目的である。そのため、本稿では、ワークフローシステム開発に利用する移動エージェントシステムや有線/無線ネットワーク



の種類などのシステム要件を定め、アプリケーションやハードウェア環境に依存するデータの大きさや、転送量あたりの転送時間を評価時の定数係数として利用している。これにより、確率統計モデルを用いた場合と比較して、たとえば 4.5 節で示した MA(Star) と RPC の通信時間の比較のように、ある特定のネットワークを利用したときの RPC に対する MA(Star) の通信時間のオーバーヘッドを見積るなどの、より具体的な議論ができたと考えている。

定量的評価は設計時のアーキテクチャ・スタイルの選択の指針となるほかに、実行時に環境に応じて振舞いを変える移動エージェントに対しても、どのように振る舞えば効率的であるかを示す有益な指針を与える。文献 3) では Network Awareness を提案し、ネットワーク資源の情報を提供するためのネットワーク・モニタを開発している。同様に文献 5) では、環境情報を提供する仕組みを持った移動エージェント基盤を提案している。文献 13) ではコンピュータ間を移動するアプリケーションに、統合的なインタフェースで環境情報を提供する環境サーバを提案している。これらの研究に加え、我々の定量的評価方法を用いることで、堅牢で効率的なシステム開発の指針を得ることができると考えている。

## 6. おわりに

本稿では、ワークフローシステム開発という具体例を通して、我々の提案する移動エージェントの定量的評価手法について述べた。従来の評価手法と比較して、ネットワーク特性を考慮して評価している点が本手法の特徴である。たとえば、MA の利点の 1 つとして通信量の削減効果があげられることが多い。しかし、通信量の削減効果は扱うデータ量の変動が大きい情報検索などの特定のアプリケーション領域に限られたものである。ワークフローシステムのように扱うデータ量の変動が少ない、より一般的なアプリケーションの場合、本稿で行った評価のように、エージェント実行コードの転送により転送量が逆に増加してしまう場合が多い。移動エージェント技術の評価するためには、転送量からの評価だけでは不十分であり、通信時間やネットワーク特性まで考慮する必要がある。

本手法はワークフローシステムに限ったものではなく、他の分散システムに対しても適用可能である。また、特定の移動エージェントシステムに限ったものでもない。本稿の実験でアプリケーション開発に用いた移動エージェントの転送プロトコルは単純なものであったが、セキュリティや、移動先の資源の検査などの機能

を考慮した場合、転送プロトコルはより複雑になるため、転送プロトコルにあった式を作成すればよい。

今回のワークフローシステムの例題では、各データの種類の中で文書ファイルのサイズが大きく、そのため、通信時間全体で見た場合の MA(Star) と RPC との差異が小さくなる傾向にあった。この点が他のアプリケーションと比較した場合のワークフローシステムの特徴である。他のアプリケーションに本手法を適用した場合、MA(Star) の実行コード転送が RPC に対するオーバーヘッドとしてより大きな割合を占めるようになるかと予測される。

今後は、さらに定量的評価方法を他のアプリケーション開発に適用し、アプリケーションの特徴と MA の適性の関係について考察することで、どのような機能・役割に対して MA を適用すべきなのかを明らかにしていきたい。

謝辞 有益なコメントをいただいたオブジェクト指向シンポジウム 2000 の査読者の方々に感謝します。

## 参考文献

- 1) Arnold, K. and Gosling, J.: *The Java Programming Language*, Second Edition, Addison-Wesley (1998).
- 2) Bricconi, G., et al.: Issues in Analyzing the Behavior of Event Dispatching Systems, *Proc. IWSSD2000*, pp.95-103 (2000).
- 3) Caripe, W., Cybenko, G., Moizumi, K. and Gray, R.: Network Awareness and Mobile Agent Systems, *IEEE Communications Magazine*, pp.44-49 (July 1998).
- 4) Carzaniga, A., Picco, G.P. and Vigna, G.: Designing Distributed Applications with Mobile Code Paradigms, *Proc. ICSE'97*, pp.22-32 (1997).
- 5) Subagyo, H., 高汐一紀: ディレクトリサービスによる移動エージェントに対する環境情報支援, 情報処理学会研究報告, 2000-OS-83, 83-13, pp.73-78 (2000).
- 6) 本位田真一, 飯島 正, 大須賀昭彦: エージェント技術, 共立出版 (1999).
- 7) Inverardi, P., Mangano, C. and Balsamo, S.: Performance Evaluation of a Software Architecture: A Case Study, *Proc. IWSSD'98*, pp.116-125 (1998).
- 8) Ismail, L. and Hagimont, D.: A Performance Evaluation of the Mobile Agent Paradigm, *Proc. OOPSLA '99*, pp.306-313 (1999).
- 9) Lange, D.B. and Oshima, M.: *Programming and Deploying Java Mobile Agents with Aglets*, Addison-Wesley (1998).

- 10) Lange, D.B. and Oshima, M.: Seven Good Reasons for Mobile Agents, *Comm. ACM*, Vol.42, No.3, pp.88–89 (Mar. 1999).
- 11) Milojicic, D., et al: *MASIF: The OMG Mobile Agent System Interoperability Facility*, LNCS 1477, pp.50–67, Springer (1998).
- 12) 中島 震, 二木厚吉: 代数アプローチによるアーキテクチャ・スタイル記述, *FOSE'98* (1998).
- 13) Nakajima T., Aizu, H., Kobayashi, M. and Shimamoto K.: Environment Server: A System Support for Adaptive Distributed Applications, *WWCA '98* (Mar. 1998).
- 14) Satoh, I.: MobileSpaces: A Framework for Building Adaptive Distributed Applications using a Hierarchical Mobile Agent Systems, *Proc. ICDCS'2000* (2000).
- 15) Tahara, Y., Ohsuga, A. and Honiden, S.: Agent System Development Method Based on Agent Patterns, *Proc. ICSE'99*, pp.356–367 (1999).
- 16) WfMC: The Workflow Reference Model, 1995.1.19 (1995).
- 17) White, J.E.: Mobile Agents, *Software Agents*, Bradshaw, M.J. (Ed.), pp.437–472, AAAI Press/The MIT Press (Apr. 1997).
- 18) Yamahira, T., Kawashima, K., Kariya, J. and Nakajima, T.: Approach for Wide Use of Agent Communication Systems, *Proc. APNOMS'99* (1999).

(平成 13 年 4 月 5 日受付)

(平成 14 年 4 月 16 日採録)

## 推薦文

モバイルエージェント技術の諸側面に関する定量的評価は、モバイルエージェント技術が実際のアプリケーションシステムの開発に真に有効か否かを判断する上で必須の課題と思われる。本論文はこの課題に正面から取り組み、新規性のある結果を得ており、推薦に足ると判断した。

(SE 研究会主査 青山 幹雄)



前田 直人(正会員)

平成 11 年早稲田大学大学院理工学研究科修士課程修了。同年 NEC 入社。現在、同社ネットワークング研究所勤務。分散ソフトウェアに関する研究に従事。



中島 震(正会員)

昭和 54 年東京大学理学部物理学科卒業。昭和 56 年東京大学大学院理学系研究科修士課程修了。同年 NEC 入社。同社研究開発グループを経て、平成 14 年法政大学経営学部教授。学術博士(東京大学)。分散ソフトウェア工学、ソフトウェアの形式仕様と検証の研究に従事。平成 13 年より科学技術振興事業団さきがけ 21「機能と構成」領域研究員を兼務。平成 13 年度山下記念研究賞受賞。本学会ソフトウェア工学研究会幹事、日本ソフトウェア科学会理事・編集委員。