

適応型ネットワークサービス実現のための モバイルエージェントによる「環境」移動方式

本田 知也^{†1}

川越 恭二^{†2}

一般にユーザは独自の「環境」を保有端末に所持し、移動先でその「環境」を使用する。しかし、この方式では転送遅延、伝播遅延、作業の非継続性、ホストダウン、モバイルコンピュータの持ち運びによるストレス等の問題が発生する。これらの問題を解決するため、本稿ではモバイルエージェントによる「環境」移動方式を提案する。提案する方式は、ユーザ独自の「環境」を移動先にモバイルエージェントにより移動、適応させる方式である。提案する方式では、ユーザの物理的位置、嗜好等のユーザ情報を利用するため、複雑なモバイルエージェント移動方式を使用することなく、簡単にモバイルエージェントを動的な環境へ適応することができる。従来の類似方式と提案する方式との比較実験、および動的な環境における提案方式の評価実験を行うことで、提案方式の有効性を検証した。

Environment Migration Method with Mobile Agents for Adaptive Networking Services

TOMOYA HONDA^{†1}

KYOJI KAWAGOE^{†2}

Users have their environment, software and their configuration in their owner computer and use the environment even in remote sites. However, in such remote accesses, some severe problems occur that include the data propagation delay, the data transmission delay, the user task discontinuousness, the host down, and stress caused by carrying computers. In order to solve these problems, we propose Environment migration method with mobile Agents for Adaptive networking Services. With the proposed method, user environment can be migrated depending on user movements and is adjustable. Moreover, the proposed method employs simple mobile agent operations, controlled by user information. With some simulations, the proposed method is shown to be effective, compared with the existing methods.

1. はじめに

最近のネットワーク技術の発展により、データ通信に用いることのできる帯域幅やサービスが急速に拡大している。また、携帯電話やPDAに代表されるような小型携帯端末の出現、低コスト化による端末の普及により、ユーザは様々な場所が必要な情報を取得し、作業が行える環境を持つことができる。

この状況の中、ユーザは、保有している計算機に独自の「環境」を構築し、その環境を用いて作業を行う。デスクトップの形態や構造、ユーザが使用しているアプリケーションおよびその設定、ユーザの作業データ等に見られるようにユーザ毎

に独自の「環境」が存在する。ユーザはその「環境」を自宅、電車、オフィスにおいて使用する。自宅ではデスクトップPC、電車ではノートPC、オフィスでは会社のPCと様々な形態で計算機を利用する。しかし、移動先毎にユーザ独自の「環境」を構築し、またそれぞれの同期を取ることはユーザに多大の負担が掛かる。これまで、この負担を軽減するため、ネットワークを利用したVNC(Virtual Network Computing)[1]などの遠隔操作ソフトウェア、ハイパネーション機能を用いてOSと共に状態を転送させる方式、移動端末に「環境」を保持し周辺と適応させる方式等が実現され、上記の問題の解決に向けて取組まれている。しかし、従来の方式では通信のレイテンシを常に伴う問題や、作業の継続性の欠如、OSのディスクイメージを転送する際に発生する転送時間等の問題が存在すると考える。

^{†1}立命館大学大学院 理工学研究科

Graduate School of Science and Engineering, Ritsumeikan Univ.

^{†2}立命館大学 理工学部

School of Science and Engineering, Ritsumeikan Univ.

そこで、上記の問題を解決するために、モバイルエージェントの自律性、移動性、状態の継続性、またユーザの情報を利用することで「環境」を移動させる方式を提案する。以降、2章では既存の「環境」移動方式をモバイルコンピューティング方式とネットワーク通信方式で分けて説明し、問題点を述べ、3章ではモバイルエージェントによる「環境」移動方式その有効性を述べる。4章では本提案方式を実現するための、各フェーズ（ユーザ情報取得、コスト計算、波状型単巡回方式）と、本方式を用いたシステム構成について述べる。5章にシミュレーションによる実験と結果、6章でまとめと今後の課題について述べる。

2. 既存の「環境」移動方式

本章では、既存の「環境」移動方式を、モバイルコンピューティング方式とネットワーク通信方式に分けて説明し、両方式の利点と問題点を述べる。更に、その問題点を現在の状況と照らし合わせ、解決すべき形態と問題に絞り込む。

2.1 モバイルコンピューティング方式

本稿では、モバイルコンピューティング方式とは、モバイル端末上に「環境」を保持し、ユーザと共に移動し、移動先で使用する方法と定義する。モバイルコンピューティング方式は、これまでも、計算機における移動透過性やサービスの透過性、作業の継続性およびその品質保証において、Mobile-IPやTCPのセッションを保つ方式等の様々な研究がなされている。[2][3]

しかし、ユーザがモバイル端末を保持し移動するには計算機を持ち運ぶため、ユーザに多大な負担が掛かる。また、負担の掛からないPDAや携帯電話では、計算機自体の性能の面で作業範囲が固定してしまうという問題がある。

2.2 ネットワーク通信方式

本稿では、ネットワーク通信方式とは、ネットワークを用い、「環境」を保持しているホストと通信を行い使用する方式と定義する。ネットワーク通信方式は、VNCなどの遠隔ソフトウェアと、「コンピュータテレポーテーション」[4]に分類できる。

前者の遠隔ソフトウェアでは、ホスト上の「環境」を移動先ですべて使用することができる。しかし、この方式には、常に通信遅延問題がある他、使用時にコネクションを確立しておかなければならないという問題もある。また、作業の継続性、ホストダウン等の問題もある。

後者の「コンピュータテレポーテーション」では、「Install once, Use any computer」の下に

研究されている方式である。ハイパネーション機能を用いOSの実行を止め、止めた状態を移動先の計算機に転送し、再開する。この方式では、「環境」との通信遅延や作業の継続性の問題は解決されている。しかし、OS自体の転送による転送時間が掛かるという問題がある。

2.3 設定形態

前述した利点および問題点より、解決すべき問題を列挙する。

- 1)ユーザは携帯端末を保持することなく、移動先において「環境」を使用する形態、つまりネットワークを介しての方式が望ましい。
- 2)またユーザの作業の継続性、「環境」との通信遅延の改善、透過性のあるサービスの実現が必要である。

3. モバイルエージェントの動作

2.3で記述した問題を解決するために、「環境」を移動先で使用するための専用のモバイルエージェントを使用する。本章では、モバイルエージェントの基本動作を述べ、モバイルエージェントを使用するにあたっての有効性を述べる。

3.1 モバイルエージェントの基本動作

提案方式におけるモバイルエージェントは、以下の基本動作(図1)を行う。

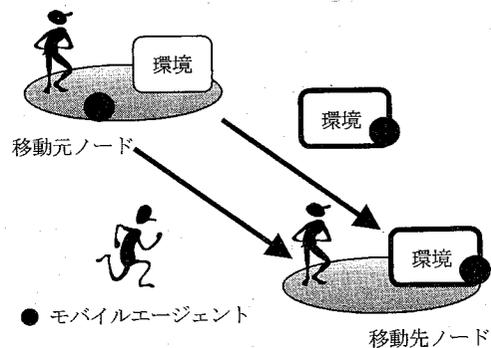


図1. モバイルエージェントの基本動作

- ①まず、管理サーバ上に常駐し、モバイルエージェントの制御を行う管理エージェント(以後、MAとする)がユーザ情報、選別し移動元ノードへ通知する。(Notification フェーズ)
- ②そして、通知を受けた移動元のモバイルエージェントが、通知を受けた情報より「環境」を選別し、「環境」を保有したモバイルエージェントである環境エージェント(以後、EAとする)を構築する。(Construction フェーズ)

③次に、ユーザの移動先ノードまたはその周辺ノードへと「環境」と共に移動する。

④移動後は、移動先でホストとしてユーザと通信を行う。

⑤その後、ノードのリソース不足や周辺トラフィックの増加に柔軟に適應する。(Searching フェーズ)

⑥なお、「環境」変更時、必要であれば移動元との同期を行う。

3.2 モバイルエージェントによる有効性

モバイルエージェントによる「環境」移動は、狭帯域のみならず、広帯域ネットワークでも、以下に示すようにその有効性は高いと考える。

1) トラフィックの軽減

移動先ノードまたはその近辺ノードに、モバイルエージェントが移動することによって、ノード内または近接した通信に範囲を局所化(図2)でき、トラフィックを軽減することができる。

2) 転送遅延・伝播遅延

従来の遠隔方式では、狭帯域での不安定な動作が転送遅延の問題となる。広帯域では伝播遅延の問題がネックとなる。前述した通信の局所化によりこの問題を解決することができる。

3) 耐故障性

移動元の故障・保障ダウンに対し、モバイルエージェントが自律的に察知することにより、ユーザの「環境」を自動的に退避することができる。

4) 周辺環境による適応

移動先ノードやその周辺ノードのリソース、トラフィック量を考慮することで、好条件な動的ノード変更、通信経路変更が可能になる。

5) 非同期実行

モバイルエージェントにより、アプリケーションの設定等に必要不可欠なデータの透過的な移動元との同期を行うことができる。

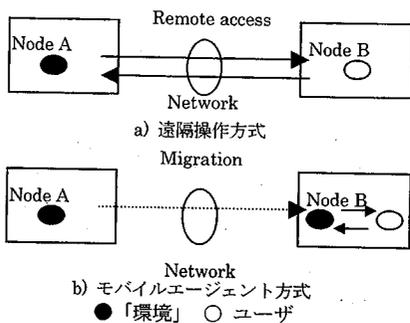


図2. 通信の局所化

4. 提案方式

本章では、3.1 で示した Notification(通知)フェーズ、Construction(構築)フェーズ、Searching(探索)フェーズについて、各フェーズ(図3)の処理の流れと、用いるコスト計算を共に述べていく。

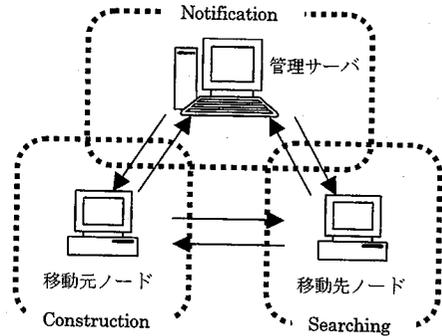


図3. システム構成と各フェーズの図

4.1 Notification(通知)

ユーザが移動後、管理サーバ上のMAが、「環境」情報を生成し、モバイルエージェントに通知するフェーズである。本方式はユーザ情報、モバイルエージェントを管理、また「環境」の退避場所として管理サーバを設ける。

4.1.1 処理の流れ

Notification フェーズの流れを以下に示す。

Step-1 ユーザが移動後、移動先ノードから管理サーバ上のMAに移動通知を行う。通知方法には、センサ、WEBなどによる複数の通知方法がある。移動通知する情報には、ユーザ認証情報、移動先ノードのIP、システムリソース情報がある。

Step-2 移動通知を受けたMAは、管理サーバ上のデータベースより、ユーザ情報を検索する。ユーザ情報には、移動元ノードIP、アプリケーション情報(名前、使用頻度、最適システムリソースと最適通信帯域)、構造、EA退避履歴がある。

Step-3 ユーザ情報より必要なアプリケーションを選択する。また以降のSearchingフェーズで必要となる探索有効コストを算出する。この二つを「環境」情報とする。

Step-4 移動元のモバイルエージェントに「環境」情報とユーザ情報を送信する。

4.1.2 探索有効コスト算出

探索有効コストは、有効移動コスト C_{EF-MV} 、有効サービスコスト C_{EF-SP} の二つであり、次式で与えられる。

$$C_{EF-MV} = a_{EF} \times S_{EA} \quad (a_{EF} \text{ は定数})$$

$$C_{EF-SP} = w_1 C_{EF-SP} + w_2 C_{EF-NW} \quad (w_1 + w_2 = 1)$$

S_{EA} は EA のサイズである。ただしアプリケーションに付随するデータは除いている。また、 C_{EF-SP} は有効処理コスト、 C_{EF-NW} は有効ネットワークコストであり、次式で与えられる。

$$C_{EF-SP} = \frac{a_{SP}}{EA_{SP}} \quad (a_{SP} \text{ は定数})$$

$$C_{EF-NW} = \frac{a_{NW}}{EA_{NW}} \quad (a_{NW} \text{ は定数})$$

EA_{SP} は EA の有効処理速度、 EA_{NW} は EA の有効帯域幅であり、次式で与えられる。

$$EA_{SP} = \sum_{i=0}^N w_{AP_i} SP_{AP_i}$$

$$EA_{NW} = \sum_{i=0}^N w_{AP_i} NW_{AP_i}$$

N および w_{AP} は必要アプリケーションの数と各使用頻度である。また、 SP_{AP} および NW_{AP} は各アプリケーションに最適な処理速度と帯域幅である。

4.2 Construction (構築)

管理サーバ上の MA より取得した「環境」情報により、「環境」を保持したモバイルエージェントである EA を構築するフェーズである。

4.2.1 処理の流れ

Construction フェーズの流れを図 4 に示す。ここで、EA の移動先への転送には、AgentSpace が提案する一括転送方式[5]を採用する。この方法は、エージェントの実行時に、必要なクラスファイルを全て移動時に転送する特徴があり、移動元との回線を切断できる利点がある。また、転送時には直列化した状態と管理情報、クラスファイルのビット列をデータ圧縮するため、転送速度の高速化が実現できる。更に、MA からのユーザ情報から、移動先ノードが最適サービスを行えるのであれば、EA に探索機能を付加することなく転送できる。

4.3 Searching(探索)

サービス品質(処理効率、通信品質)を上げるために、EA が探索有効コストを基に、最適ノード、経路を探索し、EA が動的ノード変更、動的経路変更を行うフェーズである。

```
public class Construction {
    int num = get_need_ap_num();
    for (i=0; i <= num; i++) {
        String need_ap = get_need_ap(i);
        add_cls(need_ap);
    }
    search_method();
    compression();

    public static get_need_ap_num() { ... }
    // ユーザ情報より必要なアプリケーション数を取得
    public static get_need_ap(ap_num) { ... }
    // ユーザ情報より必要なアプリケーション名を取得
    public static void add_cls(class_name) { ... }
    // モバイルエージェントにクラスファイルを付加
    public static void search_method() {
        if(j=1) { // j=1 の時は移動先ノードが最適である
            add_cls(search_name); // search_name=探索クラス
        }
        //探索クラスを付加
    }
    public static void compression() { ... }
    // 圧縮作業
}
```

図 4. Construction フェーズアルゴリズム

4.3.1 最適ノード・経路探索機能

移動先へ EA が転送された後、移動先ノード負荷や周辺のトラフィック問題で、ユーザが最適なサービスを得られない場合がある。そこで、最適ノード・経路探索を行い、EA を最適ノードに再配置する。

探索行動は、EA から探索エージェントを創出し、探索させ EA に通知する。モバイルエージェントによる探索は、様々な方式[6][7]が提案されているが、本方式は、動作を単純にするために、ユーザ情報、「環境」情報を用いることとする。

探索エージェントは最適ノード探索、経路探索の二つの機能のみのモバイルエージェントであるため軽量化されている。また、ユーザ情報より移動先ノード IP が判明している点と、移動先ノードとの通信遅延を考えると、探索範囲は限定される。これにより、移動先ノードを中心とした、探索有効コストを用いた波状型単巡回探索が適していると考えられる。

4.3.2 処理の流れ

探索エージェントの処理の流れを以下の図 5 に示す。

```

public class Search_agent {
    int i = 0, j[] = { first_cost, ea_IP }; first_node[] = { first_cost, ea_IP };
    // first_cost=初期コスト, ea_IP=移動元ノードIP
    while (i <= cost_mv) { // cost_mv=有効移動コスト  $C_{EF-MV}$ 
        Agent_move dt1 = new Agent_move();
        dt1.move();
        cost_ea = dt1.move_cost();
        i = i + cost_ea;
        if (cost_mv <= i) {
            j[1] = first_node[1];
            break; }

        int node_speed = get_speed();
        int node_net = get_net();
        int node_IP = get_IP();
        j[0] = node_speed + node_net;
        j[1] = node_IP;
        if (j[0] <= first_node[0]) {
            j[0] = first_node[0]; j[1] = first_node[1]; }
        if (j[0] <= cost_sv) break; //cost_sv=有効サービスコスト  $C_{EF-SV}$ 
    }
    call_agent(j[1]);
    public static get_speed() { ... }
    //  $a_{SP}$  サイズの処理を実行し処理時間を計測
    public static get_net() { ... }
    //  $a_{NW}$  サイズの通信を行ない通信時間を計測
    public static get_IP() { ... } // IP 取得
    public static void call_agent(IP) { ... }; // EA を再配置する
}

public class Agent_Move extends Ea_agent {
    public static void move() { ... } // 次のノードに移行
    public static move_cost() { ... }; // EA 移動に掛かるコスト算出
}

```

図5. 探索エージェントアルゴリズム

5. シミュレーション実験

4章で説明した提案方式の有効性を示すために、シミュレーションプログラムを作成し、本提案方式と類似研究との比較実験、Searching フェーズにおける、探索有効コストを用いた波状型単巡回探索の有効性検証実験を行った。シミュレーションにおける実験条件と結果を述べる。

5.1 実験1

5.1.1 実験内容

実験1では、表1の実験条件の下に、遠隔操作方式と提案方式の応答時間を計測した。この実験1では、従来の方式とモバイルエージェントを用いた方式の比較を目的とするため、Searching フェーズを付加していない。表1において、*印を付けたパラメータは、各実験で括弧の範囲で値を変化させる。また、初期通信に時間が掛かりすぎるため、コンピュータテレポーテーションを比較対象としなかった。

表1 実験条件

ネットワーク環境	
環境	10×10 ノードの格子状
実行帯域幅*	約 10[Mbps](1~100[Mbps])
ネットワーク負荷	0.1~0.3 の負荷
伝播遅延*	約 200[msec](1~200[msec])
エージェントおよび処理	
エージェントサイズ*	100[KB](1~1000[KB])
処理時間	200[msec]
通信	
通信容量	約 10[KB]

5.1.2 結果と通信回数に関する考察

変化させるパラメータを、通信回数とした時の応答時間を図6に示す。遠隔操作方式の通信には不安定性が見受けられるが、提案方式では安定した通信を行っていることが分かる。この実験ではネットワーク負荷を低く保っていたが、負荷が高くなるにつれ、有効性が明確になると予想される。

また、提案方式はエージェントの移動によって初期通信に時間が掛かるが、平均応答時間を計測したところ、通信回数が5回を超えた付近で、提案方式の応答速度が速くなる結果を得た。すなわち、使用頻度の高い「環境」を扱うには提案方式が優れていることが明らかとなった。

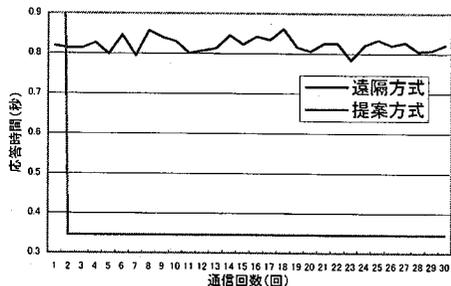


図6. 実験1-結果(回数)

5.1.3 結果とエージェントサイズに関する考察

変化させるパラメータを、エージェントサイズとした。結果として、エージェントサイズが増す毎に初期通信の時間が増加していき、初期通信以後は、通信遅延の無い安定した通信が認められた。

ここで、初期通信の時間増加という問題は、ユーザ情報による「環境」の選択と、転送前の圧縮作業により軽減される。また、初期通信の時間増加という問題よりも、EAを移動することによるトラフィックの軽減やコネクション確立の不要性という効果の方が、この問題よりも重要性が高いと予想できる。

5.1.4 結果と帯域幅に関する考察

この実験では、帯域幅の問題に絞るため、評価項目を累積応答時間とし、初期通信の問題を排除し、累積回数は10回とした。変化させるパラメータを帯域幅とした時の応答時間の変化を図7に示す。図7に示すように、すべてにおいて提案方式が良い結果を出している。この理由として、狭帯域ではリンク数を減らすことによって、安定した通信をすることができ、また転送遅延を軽減しているからである。広帯域では伝播距離を短縮することによって、伝播遅延を軽減している結果だと考える。

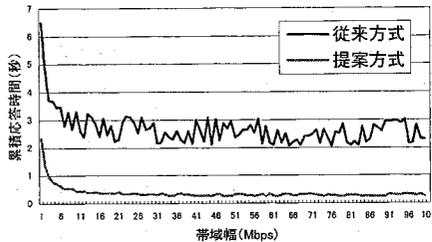


図7. 実験1-結果(帯域幅)

5.2 実験2

5.2.1 実験内容

実験2では、表2の実験条件の下に、探索機能を持つエージェントと持たないエージェントで比較実験を行い波状探索方式の有効性を検証した。Point1(0回目)、2(30回目)、3(60回目)を設け、それぞれのPointで端末負荷、ネットワーク負荷を、0.0~0.2、0.3~0.5、0.6~0.8とし、ランダムに配置した。変化させるパラメータを、エージェントサイズとし、応答時間を評価項目とした。ここで、探索機能を持たないエージェントは、ユーザ情報より移動先ノードか周辺ノードのどちらかを選択できるので、移動先か移動先ノードと周辺にいるものとの平均応答時間を結果とした。

表2 実験条件

ネットワーク環境	
環境	100×100 ノードの格子状
実行帯域幅	10[Mbps]
伝播遅延	1~50[msec](リンク数で変化)
エージェントおよび処理	
エージェントサイズ	100[KB]
探索エージェント	100[KB]
処理時間	100[msec]
通信容量	約10[KB]

5.2.2 結果と考察

図8よりPoint1で0回、Point2で1回、Point3で2回の再配置が行われている。負荷が大きくなるにつれ、再配置の瞬間にEAの移動時間が掛かり、

通信速度が落ちている。しかし、それ以外は探索機能を持つエージェントの方が、応答速度が高い。また各Pointでの平均応答時間においても、Point1で探索機能有り、無しとも0.22秒、Point2で0.29秒、0.55秒Point3で0.59秒、1.12秒であった。以上の結果より、動的環境において探索有効コストを用いた波状型単巡回探索が有効であることが分かる。

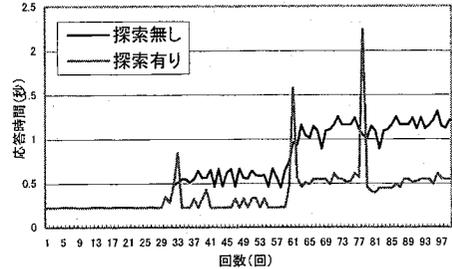


図8. 実験2-結果

6. おわりに

本稿では、適応型ネットワークサービス実現のためのモバイルエージェントによる「環境」移動方式を提案した。提案した方法は、「環境」をモバイルエージェントにより移動させる方式であり、通信遅延解消、動的環境への適応が実現可能である。

なお、今後下記の点を行う予定である。

- ・有効移動コストの閾値の決定方法
- ・周辺資源への適応およびEAの縮小化の方式
- ・連続メディアでの「環境」移動方式の検証

参考文献

- [1] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood, and Andy Hopper. Virtual Network Computing. IEEE Internet Computing, Vol. 2, pp. 33-38, 1998.
- [2] 権藤俊一, 永田智大, 岩本健史, 西尾信彦, 徳田英幸. ウェアラブルネットワークにおける移動適応型分散通信機構. 情報処理学会, DICO M シンポジウム, 鳴戸, 7, 2001.
- [3] A. Snoeren and H. Balakrishnan, "An End-to-End Approach to Host Mobility", 6th ACM/IEEE International Conference on MobiCom, August 2000.
- [4] 須崎有康. ネットワークを渡り歩けるコンピュータ. 第三回プログラミングおよび応用のシステムに関するワークショップSPA 2000, 2000年3月.
- [5] Ichiro Satoh, "MobileSpaces: A Framework for Building Adaptive Distributed Applications using a Hierarchical Mobile Agent System", Proceedings of IEEE ICDCS, pp. 161-168, IEEE Computer Society, April, 2000
- [6] 園分俊介, 清水康宏, 川越恭二. 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操作方式の提案. 情報処理学会, DICO M シンポジウム pp. 271-276, Vol. 2001, No. 7, 2001
- [7] 園分俊介, 反原清太郎, 川越恭二: 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操作方式の提案. 情報処理学会論文誌, pp. 2945-2948, Vol. 41, No 19, 2000