

モバイルエージェントを用いた ファイル検索システム方式の提案と構築

長橋 和哉† 塩澤 秀和‡ 小泉 寿男‡

東京電機大学 大学院理工学研究科
〒350-0394 埼玉県比企郡鳩山町石坂

E-mail: †kaz@itlab.k.dendai.ac.jp, ‡{shiozawa,koizumi}@k.dendai.ac.jp

あらまし コンピュータの低価格化・小型化と通信の発展により、ユビキタスコンピューティングや超分散環境と呼ばれるコンピュータ環境が整いつつある。本稿は、分散環境上におけるデバイス間のコミュニケーション方式と検索方式の提案を行う。本方式はユビキタスコンピューティング環境におけるサーバを必要としない非集中システム型を基盤としている。本方式は検索処理に評価ポイントを設け、検索時間・結果の質が悪いエージェントは淘汰され、良い検索機能を持った検索エージェントだけが残っていく検索方式である。プロトタイプを LAN で接続した PC 上に構築し、ディスク内に存在するファイル検索による実証を行い、本方式の実現の可能性を見出した。

キーワード モバイルエージェント、ファイル検索、AgentSpace、選択・淘汰

A Proposal and Its Implementation of File Search System by Mobile Agent

Kazuya NAGAHASHI†, Hidekazu SHIOZAWA‡, and Hisao KOIZUMI‡

Department of science and engineering, Graduate School of Tokyo Denki University
Shizaka, Hatoyama, Saitama, 350-0394 JAPAN

E-mail: †kaz@itlab.k.dendai.ac.jp, ‡{shiozawa,koizumi}@k.dendai.ac.jp

Abstract Environment of ubiquitous computing and super distributed computer is spread out by the downsizing of computers and the big progress of networks. This paper proposes a method of communication and file search among devices in the distributed environment. The method is based on no-centralized system that does not need servers in ubiquitous computing environment. This method adopts a concept of evaluation points in search process in which agents with poor search time and results are dismissed and agents with good search functions survive. The proposed system was implemented to search files on PCs connected by LAN, where the possibility of feasibility of the method was confirmed.

Key words Mobile Agent, File Search, Agent Space, Selection

1. はじめに

コンピュータの低価格化, 小型化は家電へコンピュータを組み込むことを加速させ, 現在, ユビキタスコンピューティングや超分散環境といわれるコンピュータ環境が進展しつつある[1]. 情報家電は1つのデバイスで1つの仕事をするように設計されているが, 多数の家電が集まりそれぞれの機能を補い1つの機能を構成するコミュニケーション方式も重要な課題になっている. このような様々な環境を持った分散コンピューティング上でデバイス同士が連携し1つの機能を提供することは, 利用者の利便性を高めることにつながる. 利用者は様々な場所で様々なコンピュータを利用するため, デバイス毎に保存されるデータは分散化する. このため, 超分散環境での情報検索には新たな課題が発生する. 動作環境が多様多様であるので適切な検索方法を実装していない可能性や, 検索依頼者が望む検索方法を実装していない可能性がある. このような課題に対して, どのデバイスがどのような情報を保存しているかを管理するためにサーバコンピュータを設ける方式がある[2][3][4]. しかしながら, サーバはデバイスを集中管理する特性上, 信頼性対策, セキュリティ対策, 安定電源の確保策が要求され, システムのメンテナンスにも配慮が要求される. 超分散化の環境が進むにつれ, サーバ方式以外の非集中的な方式が必要になると考える.

本稿は, 上記のような分散環境上におけるデバイス間のコミュニケーション方式と検索方式の提案を行う. 本方式では集中管理するためのサーバを必要としない方式でデバイス間のコミュニケーションを行う. デバイスのインターフェースに依存することなく検索を行うためモバイルエージェントを用いる. しかし, 検索命令をすごとにエージェントが移動することはネットワークオーバーヘッドを増大させる. このためエージェントの移動基準として, 検索処理に評価ポイントを設け, 検索時間, 検索結果の質が悪いエージェントは淘汰され, 良い検索機能を持ったエージェントだけが残っていく検索方式を提案する. 提案方式をLAN上に存在するPCでのファイル検索に適用し, モバイルエージェントを用いて構築した. プロトタイプを構築し, 本方式の動作を確認した.

2. デバイス間コミュニケーション

図1に, サーバ管理方式と非集中システム概念図を示す. 図中aのサーバ管理方式は, パソコンで構成されるネットワークと同様, 1つのサーバコンピュータが複数デバイスを管理するという構成である. サーバコンピュータの停止はネットワークの停止を意味し, そのため大きな信頼性が求められる. しかし, 家庭でサーバコンピュータを管理するためには技術力や経済面から考えて適当であるとはいえない. また, 超分散環境でのサーバ動作はオーバーヘッドを加速させる可能性がある. 一方, 非集中システムと言う考え方が注目されている[5].

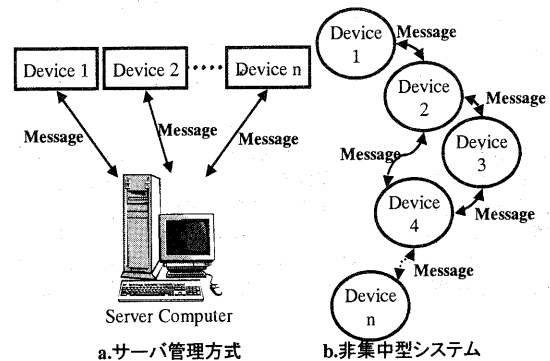


図1 サーバ管理方式と非集中システム概念図

非集中システムは生物界に存在する管理システムを参考にした協調・連携の考えであり, 各要素間の局所的な相互作用によって様式が決定される. 例えば, アリは餌を集団で餌場から巣へと運ぶが, 指揮役のアリはいない. アリは餌を見つけるとフェロモンを出しながら巣へ戻るという特徴があり, このフェロモン情報に従って他のアリも集まり, 1つの集団を成す. 非集中システムは, アリの採餌行動に見られるような特定の管理者が存在しないにも関わらず, 個々が同じ目標を持ち集団が1つの目的を成すシステムであり, サーバ方式のような管理機能を必要としない特徴がある. 図中bは, デバイス間の連携を非集中型のシステムで構築した概念を示す.

本稿では, メッセージ連携による連鎖反応機能を持った非集中システムを取り入れ, サーバコンピュータを必要としないコミュニケーション方式を提案し, この方式をファイル検索に適用した.

3. ファイル検索方式

3.1 本方式の流れ

(1) デバイス間連携方式

本方式の全体的な流れを図 2 に示す。

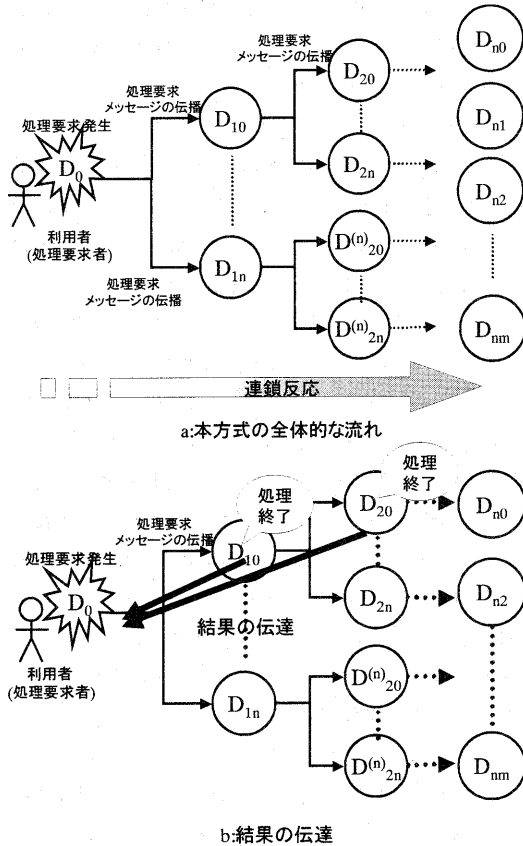


図 2 本方式の全体的な流れ

図 a において D_0 が処理要求発生地点である。利用者が D_0 に処理を命令すると、 D_0 は処理要求を D_{10} から D_{1n} に伝播し、同様に、 D_{10} は D_{20} から D_{2n} に処理要求を伝播する。各デバイスで処理をした結果は、直接処理要求発生元に伝達する。結果伝達の流れを図 b に示す。このようにして、要求を次々に伝播することで、連鎖反応を起こし各デバイスで処理し、結果を処理要求者に送信する。図 b では、 D_{10} と D_{20} で処理終了後、結果を処理要求者の元へ伝達する。

本方式では、上記の連携方式をもとにモバイルエージェントによる検索機能を実現する。検索対象は利用者が各デバイスを利用したときにデバイス内に保存した情

報である。モバイルエージェントは、上に述べた連携に沿ってデバイス間を移動する。エージェントの移動は、移動先に認証され、許可されたときのみとする。

(2) 評価ポイント比較による検索

このような連携方式を元に検索を行うと、デバイス毎に実装している検索アルゴリズムが異なるため、検索依頼者の検索要求を満たせない可能性がある。したがって、検索機能を移動エージェント化し、デバイス間に移動して検索を行う。移動して検索を行うかどうかの判断のために、検索時間、利用者が求める検索結果の質に関する評価ポイントの概念を取り入れる。評価ポイントの良い検索エージェントだけが残っていくという選択・淘汰方式を取り入れた。評価ポイント SP(Search agent evaluation Point)は次式で定義する。

$$SP = \alpha(F_s / T_s) + \beta U_{ep} \quad \dots \text{式(1)}$$

但し $\alpha + \beta = 1 (\alpha > 0, \beta > 0)$

式(1)において F_s は検索対象アイテム数、 T_s は検索にかかる時間[ms]、 U_{ep} は検索依頼者が検索結果を評価した値であり 1~5 の 5 段階で評価する。

この SP を用いて検索エージェントの選択・淘汰を行う。流れを図 3 に示す。

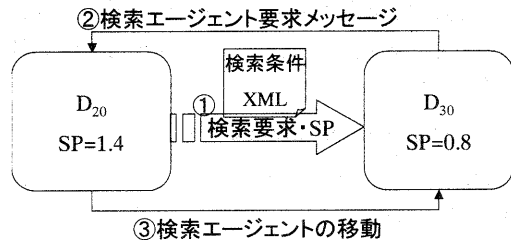


図 3 検索エージェントの移動

図において、メッセージ伝播元は D_{20} であり、メッセージ伝播先は D_{30} である。メッセージを伝播する際、 D_{20} は検索エージェントの SP 値を送信する(図中①)。 D_{30} に存在する検索エージェントの SP 値に比べて D_{20} に存在する検索エージェントの SP 値が高いときには、 D_{30} から D_{20} へ検索エージェント要求メッセージが出される(図中②)。 D_{20} 検索エージェントが D_{30} へ移動する(図中③)。

(3) 検索のフロー

検索システムのフローを図 4 に示し、検索の流れと検索エージェントの移動について述べる。

(iii)SP 値の算出を行う。

```
double getSP(long time,int fileNumber,int Usp);
```

(i)はキーワードの配列と検索対象ディレクトリを受け取り検索結果を配列として返す。(ii)は検索条件 XML と検索対象ディレクトリを受け取り検索結果を配列として返す。(iii)は検索時間と検索対象ファイル数、ユーザの評価結果から SP の値を計算する。検索条件に合うファイルを発見したときは、伝播・転送エージェントをロードし、検索結果を伝える。

③ 転送・伝播エージェント

転送・伝播エージェントは、検索依頼者が発した検索条件 XML の伝播と検索エージェントが発見したファイルを検索依頼者に送信する機能を持つ固定エージェントである。検索条件 XML 受信時に検索条件 XML に格納されている伝播上限回数値を1つ減らし、他のパソコンへ伝播する。伝播回数上限値が0であるとき、または、伝播時間の上限値を超えているときは他のコンピュータへの伝播は行わない。

3.3 エージェントの連携動作

図 6 に検索システムにおけるエージェントの連携を示す。検索依頼者が情報管理エージェントに検索を命令すると情報管理エージェントは、検索条件 XML を作成する。伝播・転送エージェントは情報管理エージェントに伝播先コンピュータリストの送信要求をし、情報管理エージェントは、コンピュータリスト XML を作成する。

伝播・転送エージェントは、コンピュータリスト XML に格納されているコンピュータに検索条件 XML を送信する。受信側のコンピュータでは伝播・転送エージェントが検索条件 XML を受け取り、検索条件 XML に格納されている SP 値と自分が保存している SP 値とを比べ、検索条件 XML に書かれている SP 値が高いときには検索エージェントが移動し選択・淘汰が生じる。

検索エージェントが検索条件を満たすファイルを発見すると伝播・転送エージェントが呼び出され検索依頼者の元へファイルを転送する。ファイル転送後、情報管理エージェントが検索依頼者に検索結果について評価を求め、その評価は検索エージェントへ送信され、新たな SP 値が求められる。

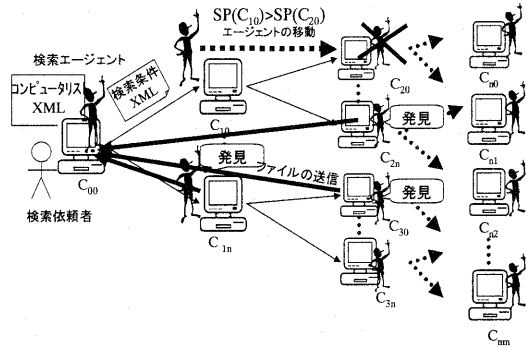


図 6 検索システムにおけるエージェントの連携

4. プロトタイプの構築

プロトタイプの構築基盤は、Windows2000 上で Java2 SDK1.3.1 を用い、モバイルエージェントの構築には AgentSpace を用いた[6]。LAN 接続の PC のディスクに保存されているドキュメントファイルを検索するエージェントを設計した。

4.1 AgentSpace の利用

本研究では Java を用いてシステムを構築し、モバイルエージェントフレームワークとしては AgentSpace を用いた。AgentSpace は、Java で構築されたモバイルエージェントフレームワークであり、モバイルエージェントのロード・セーブ、エージェントフレームワーク間を移動する機能、エージェント間通信などを提供する。AgentSpace の構成図を図 7 に示す。

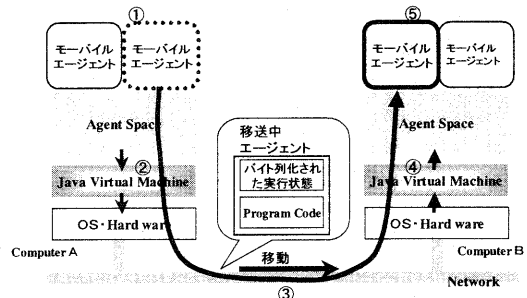


図 7 AgentSpace の構成図

AgentSpace でエージェントの移動は次のような手順で行われる。1つのエージェントが使用するクラスを JAR(Java ARchive)形式で1つのファイルにまとめる(図中①)。エージェントが ComputerA から ComputerB に移動するとき、エージェントは Java 言語の Object

Serialization 機構を利用し直列化される(図中②). 直列化されたエージェントはネットワークを介して ComputerB に移動する(図中③). 直列化されているエージェントをバイトコードに戻す(図中④). エージェントの動作を開始する(図中⑤)AgentSpace のエージェント間通信は同一ランタイムで実行されるエージェントのみの通信であり, エージェントのメソッドを呼び出すことで実現されている.

AgentSpace 以外の Java ベースのモバイルエージェントシステムは, 実行状態で転送した後, 必要なコードをオンデマンドで転送する. クラスコードの転送要求を送り, それからコードを受け取る. そのため一度に転送するデータ総量は少ないが, 最小でも2回の通信が必要になり, エージェントの転送時間を遅くする. AgentSpace は, 1 つのエージェントを構成するすべてのクラスファイルと実行状態を 1 つにまとめてからエージェントを転送するため, クラスコードを持つホストとの回線が切れてしまってもエージェントシステムを構成できる.

4.2 エージェントの構築

3 つのエージェントは AgentSpace で提供される Agent クラスを継承しており, 同じコンピュータ上に存在するときは AgentSpace の通信機能を使いエージェント間通信を行っている. しかし, AgentSpace に依存した通信機能になるのを避けるため, 通信の内容は XML を用いる方式とした. XML ドキュメントへのアクセスは Java と親和性の強い JDOM API を用いて実装した[7].

(1) 情報管理エージェント

利用者が情報管理エージェントに検索を命令したときに作成する検索条件 XML の DTD を図 8 に示す.

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<!ELEMENT SearchInformation ( performative, keyword+,
messageStepLimit, messageTimeLimit, requester, SP )>
<!ATTLIST SearchInformation uniguecode NMTOKEN #REQUIRED >
<!ELEMENT keyword ( #PCDATA )>
<!ELEMENT messageStepLimit ( #PCDATA )>
<!ELEMENT requester EMPTY >
<!ATTLIST requester ipAddress NMTOKEN #REQUIRED >
<!ATTLIST requester username NMTOKEN #REQUIRED >
<!ELEMENT messageTimeLimit ( #PCDATA )>
<!ELEMENT performative ( #PCDATA )>
<!ELEMENT SP ( #PCDATA )>
```

図 8 検索条件 XML の DTD

number 属性値はユーザが検索を命令すると乱数によって付けられる 16 桁の数値であり, 他の伝播情報と混

乱しないように付けられる. <keyword>要素には検索条件となるキーワードを複数個設定できる. メッセージを伝播する制限をステップ数と時間で設定でき, <messageStepLimit>にはメッセージが伝播してステップ数の限界を設定する. この値は伝播されるごとに 1 つずつ減らす. <messageTimeLimit>は伝播開始から何秒伝播を続けるかを設定する. 伝播を開始してから指定時間が過ぎるとそこで伝播は終了する.

(2) ファイル検索エージェント

ファイル検索エージェントのクラス構成を図 9 に示す. SearchAgent クラスは Agent クラスを継承しており AgentSpace 間での移動が可能である. AgentCommunication クラスと SearchXML クラスは他のエージェントとの通信を XML を用いて行う. SP クラスは SP を求める機能である. FileSearchAbstract クラスは抽象クラスであり, このクラスを継承して様々な検索方式をもつエージェントを実装できる.

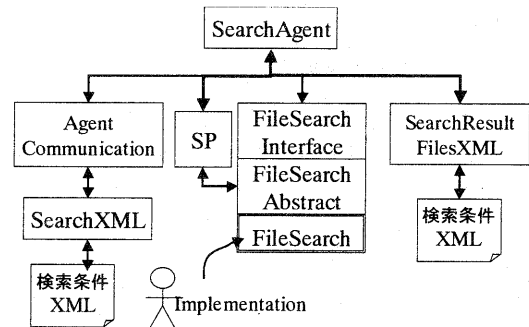


図 9 ファイル検索エージェントのクラス構成

選択・淘汰を起こさせるため3種類用意し, FileSearch クラスを実装した. 検索の方式は以下を以下に示す.

- ① Windows の Explorer の検索機能を用いた方式
- ② コマンドプロンプトで提供される find を用いた方式
- ③ ファイル名だけを対象にし検索する方式

(3) 伝播・転送エージェント

伝播・転送エージェントのクラス構成を図 10 に示す. SelectTransferAgent クラスは Agent クラスを継承しており, AgentSpace 上で動作する. 伝播・転送エージェントがロードされると Ftpd クラスが動作し, FTP によるファイルの受信を受け付けるようになる. 他の伝播・転送エージェントから検索条件 XML が転送されると,

SearchXML クラスが解析し、AgentCommunication クラスを介して検索エージェントに検索条件を送信する。

検索エージェントが条件を満たすファイルを発見した時、伝播・転送エージェントは FtpClient クラスを使い検索依頼者の PC にファイルを転送する。

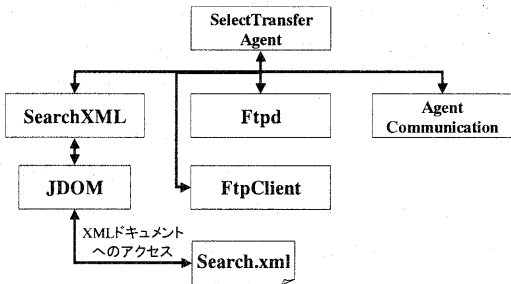


図 10 伝播・転送エージェントのクラス構成

5. 実証と考察

5.1 実証のための検索システム

実証環境を図 11 に示す。LAN 接続の PC のディスクに保存されているドキュメントファイルを検索するエージェントである。

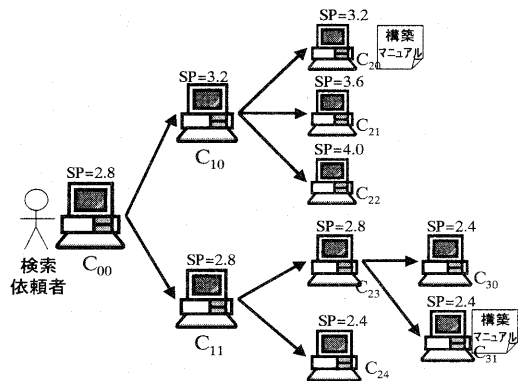


図 11 実証環境コンピュータの配置

100BASE の LAN で接続された PC, 10 台に本システムをインストールした。各 PC には 100KB のテキストファイル 50 個を保存し、図中 C₂₀ と C₃₁ だけは「SOAP-UDDI-WSDL Web サービス環境構築マニュアル」という名前のドキュメントファイルを保存した。表 1 に情報管理エージェントに設定した伝播先コンピュータと、各検索エージェントの SP 値を示す。

表 1 情報管理エージェントの設定値と SP 値

コンピュータ名	情報管理エージェントの設定値			SP値
C00	C10	C11		2.8
C10	C20	C21	C22	3.2
C11	C23	C24		2.8
C20		設定なし		3.2
C21		設定なし		3.6
C22		設定なし		4.0
C23	C30	C31		2.8
C24		設定なし		2.4
C30		設定なし		2.4
C31		設定なし		2.4

5.2 実証結果

(1) 検索命令メッセージの伝播

図 12 に示す情報管理エージェント画面に、検索条件キーワードとして「Web サービス環境構築」、伝播上限回数を「5」とした。

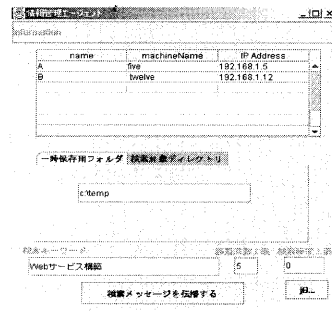


図 12 情報管理エージェントの画面

結果、情報管理エージェントによって図 13 に示す検索条件 XML が生成された。

```
<?xml version="1.0" encoding="SHIFT_JIS" ?>
<SearchInformation uniquecode="7507841648978039">
<messageTimeLimit>0</messageTimeLimit>
<performative>search</performative>
<requester username="kaz" ipAddress="192.168.1.8" />
<keyword>Web サービス環境構築</keyword>
<messageStepLimit>4</messageStepLimit>
</SearchInformation>
```

図 13 検索条件 XML

D₀₀ の情報管理エージェントが管理している IP アドレスを元に、伝播・転送エージェントから C₁₀ と C₁₁ への検索条件 XML の伝播が見られた。同様に、C₁₀ から、C₂₀, C₂₁, C₂₂ への伝播が見られ、C₁₁ から C₂₃ と C₂₄ への伝播、C₂₃ から C₃₀ と C₃₁ への伝播が見られた。また、SP 値により検索エージェントの移動は、C₁₁ から C₂₄ への移動と、C₂₃ から C₃₀ と C₃₁ への移動が見られた。

(2) 検索処理とファイルの転送

検索条件 XML を受け取った伝播・転送エージェント

は、検索エージェントに検索条件を伝え、各 PC で検索処理が見られた。検索の結果 C₂₀と C₃₁から検索条件を満たすファイルが発見され、伝播・転送エージェントにより検索依頼者(C₀₀)への転送が行われた。以下に、転送結果の画面の示す。検索依頼者が検索命令を出してから検索結果を得るまでの時間は 12 秒であった。

FileName	Owner	IP Address	Evaluation
SOAP-UDDI-W	齋藤直樹	192.168.1.15	4
SOAP-UDDI-W...	asugimo	192.168.1.18	3

図 14 転送結果の画面

(3) SP の α , β 値を変化させたときの淘汰・選択

SP 値の定義として α と β の比率を変えて、本システムで検索を行った。 α を 0.6 以上にすると検索のスピードが重視され、検索スピードの速いエージェントが選択され残った。また、 β を 0.6 以上にすると利用者からの評価値が重視され、利用者の好みが反映された検索エージェントが残った。

検索エージェントが淘汰される判断基準として SP 値を用いているが、SP 値を求める際に使われる検索時間は検索を行うコンピュータの能力に依存してしまう。よって、移動元コンピュータで SP 値が高くても、移動先コンピュータでは SP 値が高くなり、必要ない検索エージェントの移動が生じてしまうことがあった。

5.3 考察

(1) 情報管理エージェントの伝播先選択機能

今回構築したプロトタイプでは、情報管理エージェントが知りうるすべての伝播先情報を伝播・転送エージェントに渡している。情報管理エージェントに伝播先にどのようなファイルが多く保存されているかという知識を与え、高確率で検索結果が得られる伝播先を優先的に選択させることでネットワーク負荷、各デバイスでの処理時間を軽減することが可能である。

(2) 伝播・転送エージェントの受信機能

各コンピュータが同じ内容のファイルをディスクに保存している状態で検索を行うと、検索依頼者の元に多数の同一ファイルが転送されてしまうことがあり、ネットワークの負荷が増大する。情報提供先の伝播・転送エージェ

ントが検索依頼者にファイルを転送する際に転送するファイルの名前やサイズ、日付情報を送信し、検索依頼者側の伝播・転送エージェントが受信するか、しないかを判断する機能を設ける。

(3) 検索エージェントの環境認識

検索エージェントは、3.2 章で示したインターフェースを実装すれば構築が可能であるが、動作する OS により使えるコマンドや機能が異なるため、環境を判断するためのプログラムを書く必要があり、実装者の負担が多くなってしまふ。よって、検索エージェントに自分のおかれている OS やネットワークの状態を感知する機能を設け、この情報からより良い検索方式を選択するための知識を持たせる。

6. まとめと今後の課題

本稿では、モバイルエージェントを用いたファイル検索システム方式の提案を行った。エージェント機能をユーザとの対話・情報管理の機能、検索機能、メッセージ伝播、転送機能、の 3 つに分割することにより、超分散環境を含む様々な環境に適應できる検索システムを提案し、LAN 上の PC 環境にプロトタイプを構築してその動作を実証した。

今後は組み込み PDA や携帯電話上での実装を行い、本方式の検証を行っていく。また、検索エージェントへ遺伝的アルゴリズムを適用し、ユーザの評価だけで検索エージェントの一部を突然変異させ検索効率の良いエージェントを作り出すことを目指す。

参考文献

- [1] Mark Weiser, "Hot Topics: Ubiquitous Computing" IEEE Computer, October 1993
- [2] 美濃導彦, 中村素典, "情報家電の考え方", 情報処理学会誌 vol42, no11, pp. 1049-1054, Nov, 2001
- [3] NTT データ Java 研究会, "Java がもたらす近未来のネットワーク技術 Jini って何だ?", (株)カッタシステム, 東京, 1999
- [4] Chih-Lin Hu, Wen-Shyen E.Chen, "A Mobile Agent-Based Active Network Architecture", ICPADS'00, pp.445-452, Iwate, Japan, July, 2000.
- [5] 山本順他, "非集中システム", コロナ社, 東京, 2001
- [6] 佐藤 一郎, "AgentSpace モーバイルエージェントシステム", 日本ソフトウェア科学会 Workshop on Multi Agent and Cooperative Computation, Dec, 1998
- [7] JDOM, <http://jdom.org/>