

M-68

リソース配分を考慮したエージェント移動制御

Agent Migration Control Considering Resource Distribution

川上 憲治†
Kenji Kawakami広重 一仁†
Kazuhiro Hiroshige吉岡 信和‡
Nobukazu Yoshioka本位田 真一*
Shinichi Honiden

1. はじめに

近年、携帯電話や PDA などの携帯端末が広く普及し、利用場所を選ばずに、様々なアプリケーションを利用することが可能となった。しかしながら、携帯端末はバッテリーの持続時間や小型化のため、アプリケーションで利用できる計算リソース(CPU、メモリ、通信帯域幅等)に限られる。このため、デスクトップ型コンピュータなどの固定端末が利用するような、リソースを多く使うアプリケーションを携帯端末で利用した場合、リソース不足や過負荷などにより十分な性能が得られない場合がある。

このような課題を解決するためには、アプリケーションが他の端末の豊富なリソースを利用でき、また、計算リソースや通信環境の変化にも適応できる機構が必要となる。筆者らは、その機構を実現する手法として、モバイルエージェントを基盤とするミドルウェア(MA-based Middleware)を提案した [1]-[3]。提案方式は、端末間で交渉を行い、最適な端末へモバイルエージェントを移動させ、他の端末のリソースを利用することで処理能力の不足を補うことを可能とする。

本稿では、提案方式の有効性を計算機シミュレーションで明らかにする。

2. 提案システムの概要

筆者らが提案したシステムは、モバイルエージェント(以下 MA と略す)を基盤とするミドルウェア(MA-based Middleware)であり、アプリケーションから独立した機構で MA の移動を制御する。図 1 に提案システムの構成を示す。提案する移動制御方式は、図 2 に示すようにコーディネータが MA からの交渉要求(①)を受けた後、他端末のコーディネータと移動先端末を決定するための交渉を行う(②)。交渉終了後、コーディネータは MA に対して移動要求を行う(③)。

提案方式では、あらかじめ登録された移動条件を満たす環境変化(例えばリソース割当量の変化等)がノード内に生じた場合に、それをトリガとして移動先決定のための交渉を開始することを基本とする(提案方式 1)[2]。さらに、リソース割当量の豊富な端末から他の端末に通知を行うという仕組みを追加し、一層の効率化を図れる方法を提案した(提案方式 2)[3]。交渉は各端末内のコーディネータによって実行される。交渉開始の要求を受けたコーディネータは、各ノードの環境情報を収集し、MA 内に定義された評価式をもとに移動先を決定する。なお、提案方式では、交渉に用いるプロトコルに複数のエージェント間のタスク割り当てによく利用されている"契約ネットプロトコル"を用いている。

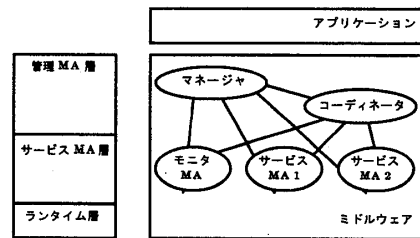


図 1 システム構成

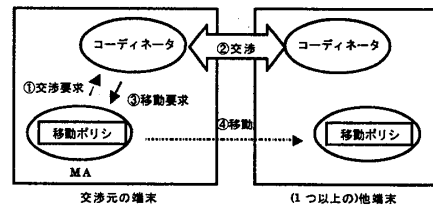


図 2 提案する移動制御方式の概要

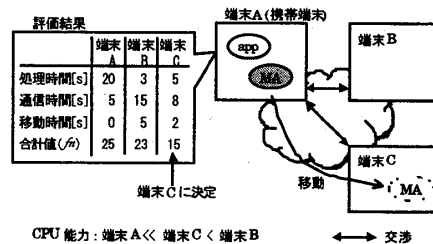


図 3 提案方式の動作例

交渉に用いる評価式 fn (n は端末を表す添え字)の一例を、以下に示す。

$$fn = \text{MAの実行時間の推定値} \\ + \text{MAのデータ通信時間の推定値} \\ + \text{MAの移動時間の推定値} \quad (1)$$

評価式(1)の各項の値は、各端末のリソース割当量等のパラメータをもとに算出される。例えば、交渉により図 3 に示す評価結果が得られた場合、MA は fn が最小となる端末 C に移動する。この例の場合、移動先の端末は単に CPU 能力が高いだけではなく、アプリケーションの処理時間を最小とるように選択される。

3. 計算機シミュレーション

3.1 シミュレーションモデル

シミュレーションモデルは、ネットワーク上に PDA が 1 台、ノート PC が 1 台、PC サーバが 2 台接続されており、各パラメータの値が表 1 で与えられる環境を想定する。各端末の使用リソース(負荷)は時間的に変動する。したがって、MA が使用できるリソースもそれに応じて変動する。負荷変動モデルとして、各端末の総リソースを 1/10 に量子

† 日本テレコム株式会社, Japan Telecom

‡ 国立情報学研究所, National Institute of Informatics

* 東京大学, The University of Tokyo

化したリソース単位で負荷を変動させる、ただし、時間毎の負荷の増減は量子化単位及び増減なしの3状態(±量子化単位, ±0)に制限し、状態変化の確率をそれぞれ 2/5(増加), 2/5(減少), 1/5(変化なし)とする。例えば、PC サーバの場合、リソースの量子化単位は 2 であり、負荷変動は、10→8(減少)→8(変化なし)→10(増加)→12(増加)のように変化する。この環境において、PDA 上で実行されているアプリケーションに着目し、その処理時間を短縮するように MA の移動を制御する。また、MA の実行時間は、タスクサイズ÷リソース割当量で導出可能なモデルを仮定する。例えば、タスクサイズが 30000 の MA を半分実行した時点で端末のリソース割当量が 10 から 12 に変動したとすると、MA の実行時間は $15000 \div 10 + 15000 \div 12 = 2750[\text{ms}]$ となる。

ところで、表 1 中の移動条件・通知条件は交渉開始のトリガを決める閾値であり、MA の滞する端末のリソース割当量が移動条件(ここでは 2 とする)よりも小さくなった場合、移動のための交渉が開始される。また、端末のリソース割当量が通知条件(ここでは 18 とする。この例ではノート PC(リソース割当量は最大 10)からの通知は行わない)以上の場合、他端末に移動の交渉を開始するように伝える。

シミュレーションは、MA がアプリケーションからのリクエストを受けて処理を実行し、レスポンスを返すまでを 1 回の処理として、これを 100 回繰り返して行う。なお、各処理の途中での移動はできないものとする。

表 1 計算機シミュレーション諸元

パラメータ	値
エージェントのタスクサイズ [ms]	30000
PC サーバのリソース割当量の最大値	20
ノート PC のリソース割当量の最大値	10
PDA のリソース割当量の最大値	1
メッセージ転送時間 [ms]	10
エージェントのコード転送時間 [ms]	64
エージェント移動時のオーバーヘッド [ms]	1000
移動先決定のための交渉時間 [ms]	1000
エージェントの移動条件	2
他端末への通知条件	18
エージェントの実行回数 [回]	100
端末のリソース割当量の変動間隔 [ms]	5000

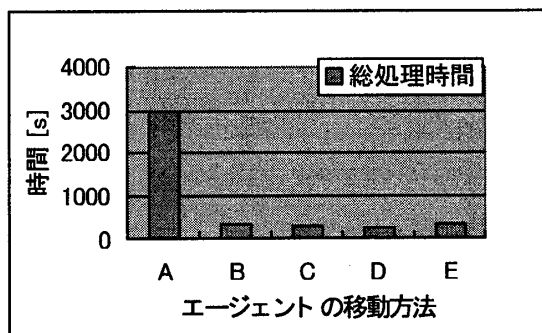


図 4 各移動方法における処理完了時間

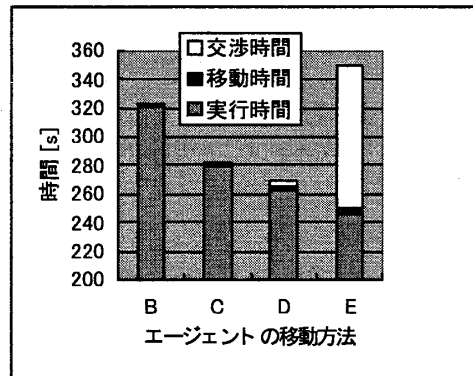


図 5 各移動方法における処理時間の内訳

3.2 シミュレーション結果

処理完了時間についてのシミュレーション結果を図 4 に示す。ここで、A～E は MA の移動方法を示しており、(A) PDA で実行し続けた場合(移動しない場合)、(B) 最初の 1 回だけ交渉・移動しそれ以降は移動しない場合、(C) 移動条件のみをトリガにして交渉を開始する場合(提案方式 1)、(D) 移動条件+通知条件をトリガとして交渉を開始する場合(提案方式 2)、(E) 定期的に交渉を行う場合、である。図より、MA が移動する場合(B～E)の処理が早く完了していることが分かる。

次に、B～E の各方法について処理時間の内訳を図 5 に示す。図より、定期的に交渉する E は、処理の実行時間を大幅に減らすことはできるが、交渉の回数が増えるため、総処理時間が長くなる。これに対して、提案方式を用いた場合(C, D)には、交渉回数の増加を低く抑えつつ、実行時間の短縮を実現できる。特に提案方式 2 は、他端末のリソースを有効に活用できることから、処理時間を大幅に低減できることが分かる。

4. まとめ

本稿では、提案したエージェント移動制御方式の適用効果を計算機シミュレーションで明らかにした。提案方式は、適切なタイミングでエージェントの移動制御が実現できることから、交渉回数の増加による時間のロスを大幅に抑えることが可能となり、処理時間を大幅に短縮できる。

参考文献

- [1] K. Hiroshige, K. Kawakami, H. Sasaki, Y. Okataku, and S. Honiden, "Mobile Agent-Based Middleware for Mobile Terminals," <http://computer.org/dsonline/0107/features/hir0107.htm>
- [2] 川上憲治, 広重一仁, 佐々木宏, 岡宅泰邦, 本位田真一, "モバイル環境向けエージェント移動制御," 情報処理学会 MBL 研究報告, 2002 年 3 月。
- [3] 川上憲治, 広重一仁, 吉岡信和, 本位田真一, "モバイル環境向けエージェント移動制御 (その 2)," 情報処理学会 MBL 研究報告, 2002 年 5 月。