

仲介エージェントを用いたモバイル環境におけるサービスのアクセス手法

3 J-7

黄 長華 大木 幹雄 坂本 康治

日本工業大学

1. はじめに

モバイルクライアント-サーバ間の通信を減らすような仕組みを実現したエージェント通信システム^[1]が開発されている。しかし、これらのシステムにおいては、それぞれのサービスが独立しており、ユーザは複数のサーバへサービスをアクセスする場合には、これらのサーバ間で頻繁に接続したり、問合せしたりしなければならない。また、すべてのメッセージがモバイルプロキシサーバ（メッセージゲートウェイとも呼ばれる）によって蓄積交換されるため、モバイルプロキシサーバにおけるトランザクションの処理負担が増加してしまう。

本研究は、これらの問題に対して仲介エージェントを導入することによりモバイルコンピューティングへのアクセス効率を改善することを目的とする。

2. 仲介エージェントの導入

従来の Client/Server システムを三階層に分けたものとして Client/Agent/Server モデルがある。このモデルはサーバ側で従来クライアントが行っていた処理の一部をエージェントに代行させて、クライアントとの間で、アプリケーション毎にあらかじめ決められたメッセージのやり取りを行う。しかし、このモデルにおいては、一つのエージェントは一種類のサービスしか提供できない^[1]。

本研究では、モバイルプロキシサーバと複数のサーバ間に仲介エージェントを複数配置する方式をとる。

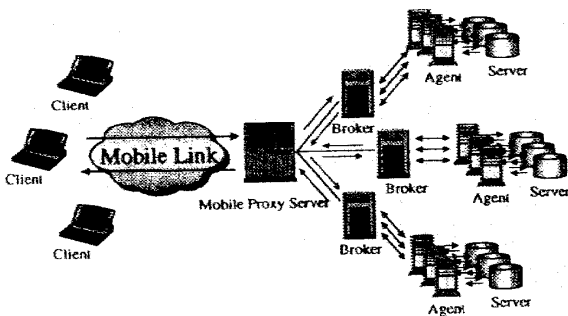


図1 仲介エージェントの導入

ユーザからの処理委託は仲介エージェントに伝えられ、仲介エージェントは複数のサーバへサービスをリクエストする。仲介エージェントはユーザ側から見ると、サーバであるが、サーバ側から見ると、クライ

アントである。システムの構成を図1に示す。

このような方式をとることにより、一つのエージェントは多数のサービスを提供することが可能となる。また、ユーザが複数のサーバへサービスをアクセスする場合には、携帯端末とモバイルプロキシサーバ間の送受信回数が最低2回で済むようになる。

3. 仲介エージェントの機能構成

仲介エージェントはメッセージサブシステムとリクエストサブシステムから構成される。サービスの位置情報およびアクセスする時に必要なパラメータがサービスアクセス情報DBに登録され、リクエストサブシステムによって管理される。図2に仲介エージェントの構成を示す。

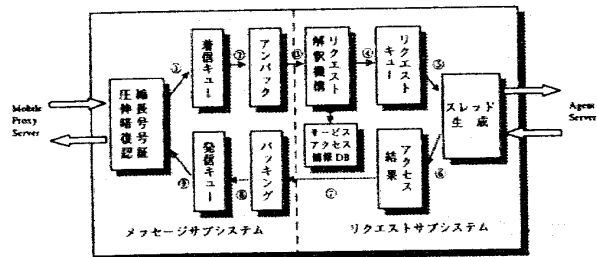


図2 仲介エージェントの構成

仲介エージェントにおけるメッセージ処理の流れは次の通りである。

- (1) 入力メッセージに対してそれが暗号化・圧縮されている場合には、伸長、複合化し、メッセージ認証を行い、着信キューにキューイングする。(図中の①)
- (2) 着信イベントを発生して、メッセージをアンバックする。(図中の②)
- (3) メッセージの中にサービスリクエストを含めばそれを解釈する。(図中の③)
- (4) リクエストをソートされ、リクエストキューにキューイングする。(図中の④)
- (5) リクエストキューの先頭から処理可能最大数までのリクエストに対しスレッドを生成して、リクエストの処理を行う。(図中の⑤)
- (6) リクエストの処理を終え、アクセス結果をもとにして出力メッセージを作成する。(図中の⑥、⑦)
- (7) 出力メッセージをバッキングされ、発信キューにキューイングする。(図中の⑧)
- (8) 最後に必要に応じて暗号化・圧縮し、メッセージを出力する。(図中の⑨)

サーバに障害が発生した時等に、メッセージの処理が無限ループに入り込んで終わらない場合も考えられる。そこで、仲介エージェントは各メッセージに対してタイムアウト時間を設定する。タイムアウトしたメッセージが自動的に削除される

4. アクセス手法の考察

4.1 サービスのアクセス手順

サービスのアクセスが以下の手順で行われる。

(1) Request: ユーザからの処理委託をメッセージとしてモバイルプロキシサーバに送る。

(2) Dispatch: 対応する仲介エージェントにメッセージをディスパッチする。

(3) Access: 仲介エージェントがメッセージを受け取り、サービスのアクセスを行う。

(4) Response: アクセスの結果を応答メッセージとしてモバイルプロキシサーバに送る。

(5) Deliver: 応答メッセージをユーザに伝達する。

サービスをアクセスする時には、ユーザの処理委託は仲介エージェントに正しく伝えられなければならない。ユーザの処理委託は次の三つの要素から構成される。①ユーザに関する情報/ユーザID、パスワード、ユーザ氏名及び連絡先の情報などである。②サービスの要求条件/ユーザの要求に関する情報をパラメータで表わしたものであり、これについて4.2以降で詳しく述べる。③リクエストの内容/アプリケーション毎に違うが、例えば、情報検索の場合には、SQL文に相当するものである。

4.2 サービスの要求条件

ユーザの処理委託を記述する時には、次のような要求条件が必要とされる。本研究では、これらの要求条件を明示的に記述し、リクエストのパラメータとして仲介エージェントに送るものとする。仲介エージェントの解釈機構はこれらのパラメータを評価実行する。

(1) サービスのアクセス場所

頻繁にアクセスするサーバの範囲を指定するものである。サービスのアクセス場所は必ず一箇所に限定するとは限らない。情報検索やショッピングサービスのような場合には、アクセスするサーバの範囲が必要とされる。

(2) アクセス制限時間

リクエストから応答結果を伝達するまでの制限時間である。指定した時間内にサービスのアクセスが終わらない場合もある。このような場合、サービスのアクセスを継続するか無効するか選定する必要がある。

(3) リクエストの優先度

ユーザのリクエストの処理優先度である。優先度の

高いものほどサービスのコストも高くなる。

(4) アクセス結果の報告方法

サービスのアクセス結果をユーザに逐一報告するかあるいはまとめて一回報告するかの指定である。

(5) 例外への対処方法

指定したサーバでのアクセスが失敗した場合、同種サービスを推薦してくれるかあるいは動的に判断してリクエストを他のサーバにまわすかの指定である。

4.3 ユーザの履歴情報

サービスのアクセス場所を明確に指定していない場合(要求条件(1))あるいは指定したサーバでのアクセスが失敗した場合(要求条件(5))、最も利用頻度の高いかつ正答率の大きいサーバの中から次候補を選んで順にアクセスすることが望ましい。

そこで、ユーザの履歴情報を

$$U = \{S_1(k_1, n_1), S_2(k_2, n_2), \dots, S_m(k_m, n_m)\}$$

と定義し利用するものとする。ただし、 n_i はサーバ S_i に対してリクエストを請求した総回数である。 k_i はサーバ S_i から応答が正しく得られた総回数である。これらの履歴情報を用いて、階層構造分析法(AHP)よりサーバ S_i の選好度 p_i を次のように定義できる。

$$p_i = w \cdot \frac{n_i}{\sum_{j=1}^m n_j} + (1-w) \cdot \frac{k_i/n_i}{\sum_{j=1}^m k_j/n_j} \quad (1)$$

$$\left[\begin{array}{l} w: \text{利用頻度のウエイト, } 1-w: \text{正答率のウエイト} \\ (\text{サーバの利用頻度と正答率を近似的に独立とみなす}) \\ k_i/n_i: \text{サーバ } S_i \text{ における正答率} \end{array} \right]$$

ユーザが一々サーバを指定することなく、仲介エージェントにより式(1)で定義した選好度の高い次候補サーバへ自動的にアクセスすることができる。

5. 終わりに

本稿では、仲介エージェントを用いたモバイルコンピューティング環境におけるサービスのアクセス手法を提案し、さらに本手法に関する問題点及び対策について考察した。現在、モバイル図書発注サービスの実装を行っており、今後、システムの実験を行って本手法の有効性を確認してゆく予定である。

参考文献

- [1] Oracle Corporation Japan: Oracle Mobile Agents アプリケーション開発者ガイド、Oracle Mobile Agents のマニュアル、1996
- [2] 松下、隆、渡辺、水野: エージェントを用いたショッピングサービスのモデル化について、情報処理学会研究報告、Vol.96, No.MBL-2, pp.13-18, 1997