

WebSocket を用いたエージェントの移動プロトコルの提案

東野 正幸*, 高橋 健一, 川村 尚生, 菅原 一孔
(鳥取大学大学院 工学研究科 情報エレクトロニクス専攻)

1. はじめに

分散システムの構築技術としてモバイルエージェントが注目されている。モバイルエージェントとはネットワークで接続された計算機間を移動できる自律的なプログラムである。

分散システムのネットワーク基盤としてインターネットを利用する場合、ファイアウォールによって、任意のプロトコルによる計算機間の通信ができない場合がある。しかしながら、ファイアウォールの多くは、計算機の利用者が WWW を利用できるように、HTTP による通信は許可している。ただし、HTTP はリクエスト/レスポンス型のプロトコルであることから、セキュリティを強化する目的で、ファイアウォールの内側からの HTTP リクエストは許可するが、外側からの HTTP リクエストは許可しない。

このようなインターネット環境において、モバイルエージェントが計算機間を双方向に移動するためには、モバイルエージェントの移動プロトコルを HTTP でトンネリングすることと、Comet⁽¹⁾に代表される HTTP 上での双方向通信技術が必要となる。

Comet を用いて双方向通信を行なう場合、送信と受信の通信路を別々に作成する必要があり、TCP を直接に用いた双方向通信と比べて、2 倍の通信路に関する資源（ポート番号、スレッド数、メモリ、など）が必要となる。これは、大規模な分散システムでは大きな負担となる。また、ポーリング方式による Comet では、通信路の疎通確認のために冗長な HTTP ヘッダが送受信されることや、Comet の実装そのものが複雑であるといったことを受けて、近年では、これらの Comet の課題を解決するような、WWW における軽量な双方向通信の技術規格である WebSocket⁽²⁾が注目されている。

WebSocket の技術目標は、Web ブラウザベースのアプリケーションが、複数の HTTP 接続を開くことなく、少ない遅延時間で Web サーバとの双方向通信を実現することである。一方、モバイルエージェントの移動プロトコルの技術目標は、省資源で高速な双方向移動を実現することである。これらの技術目標が、省資源、高速、双方向通信という点で一致していることから、モバイルエージェントの移動プロトコルとして WebSocket を用いることが有効である。

そこで本稿では、WebSocket を用いたモバイルエージェントの移動プロトコルを提案する。提案プロトコルでは、インターネット上でエージェントが双方向移動可能な

だけでなく、インターネットにおける非均質な通信速度の考慮や、WebSocket の低い通信遅延を生かした柔軟なデータ転送制御を行なうことで、エージェントの移動におけるターンアラウンドタイムやデータ転送量の削減を可能にする。

2. モバイルエージェントの概要

図 1 にモバイルエージェントの概要を示す。エージェントは、エージェント実行環境で動作し、ネットワークを介して別のエージェント実行環境へ移動できる。

エージェントの内部構造は、コード領域と実行時状態領域から構成されている⁽³⁾。コード領域には、エージェントが処理するタスクが記述された複数のコードが格納されている。実行時状態領域には、これらのタスクの処理状態に関する情報が格納されている。

このようなエージェントが、エージェント実行環境間を移動しながら動作するためには、実行時状態領域と複数のコードを、移動先のエージェント実行環境へ転送する必要がある。そこで、WebSocket を用いて実行時状態領域と複数のコードを転送する移動プロトコルを提案する。

3. 双方向通信路の維持と共有

エージェントがエージェント実行環境間を双方向に移動するためには、エージェント実行環境間で双方向通信可能な通信路を維持しておく必要がある。WebSocket では、1 つの通信路を用いて複数の双方向通信を行なうことで、通信路に関する資源を節約できる。そこで、提案プロトコルでは、2 点間のエージェント実行環境間で WebSocket の通信路を 1 つだけ維持する。ファイアウォールが設置されている場合は、ファイアウォールの外側から通信路を作成することができないため、ファイアウォールの内側から作成した通信路を維持する必要がある。ファイアウォールが設置されていない場合は、資源節約の観点から、先に作成された通信路を維持する。

1 つの通信路を共有して複数のエージェントが同時に移動する場合、エージェントの実行時状態やコードを一意に識別できなければならない。なぜなら、エージェントの移動処理は、実行時状態とコードの転送という複数の処理から構成されるため、エージェントの移動中に通信路が切断された場合に備えて、トランザクション処理が必要となるからである。そこで提案プロトコルでは、実行時状態に UUID(Universally Unique Identifier)を付与することで一意に識別する。コードは、そのバイナリデータのハッシュ値を用いて一意に識別する。

4. エージェント移動プロトコル

エージェントの移動プロトコルを WebSocket のサブプロトコルとして定義する。WebSocket ではフレームのペイロード部にサブプロトコルのフレームを定義できる。図2に提案プロトコルのフレーム構造を示す。提案プロトコルのフレームはヘッダ部とペイロード部から構成される。ヘッダ部には命令が1つ格納される。ペイロード部にはヘッダ部に格納された命令に応じた内容が格納される。命令は5種類あり、それぞれを以下に示す。

- 移動手続開始：エージェントの移動手続を開始する。ペイロード部には、1つのエージェントの実行時状態と、コードのハッシュ値の配列が格納される。
- コード転送要求：コードの転送を要求する。ペイロード部にはコードのハッシュ値の配列が格納される。
- コード転送応答：コード転送要求により得られたコードのハッシュ値に対応するコードを転送する。ペイロード部には要求されたコードの配列が格納される。
- コード転送キャンセル要求：先に送信したコード転送要求のキャンセルを要求する。ペイロード部にはコードのハッシュ値の配列が格納される。
- 移動手続終了：エージェントの移動手続を終了する。

図3、図4、図5に提案プロトコルの実行例を示す。図3は、エージェントの移動手続開始から、コード転送を経て、移動手続終了するまでの流れである。図4は、移動先にコードが既に存在する場合に、コード転送を行なうことなくエージェントの移動手続が完了する場合の流れである。図5は、一旦コード転送要求を送信したが、後からそれをキャンセルする場合の例である。これにより、例えば、同じコードを必要とする複数のエージェントが、複数のエージェント実行環境から1つのエージェント実行環境へ同時に移動手続開始した場合に、通信速度の遅い移動元からのコード転送はキャンセルし、通信速度の速い移動元から優先的にコードを転送することで、ターンアラウンドタイムやコードの転送量を削減することが可能となる。

5. おわりに

本稿では、HTTPを拡張したWWWにおける双方向通信の技術規格であるWebSocketを用いたエージェントの移動プロトコルを提案した。提案プロトコルにより、ファイアウォールが設置されている場合であっても、WWWの利用が許可されていれば、HTTPを用いた場合に比べて、短いターンアラウンドタイムと少ない転送量でエージェントを計算機間で双方向に移動できる。

今後は、提案プロトコルの実装と動作検証を行ない、HTTPを用いた場合に対する有効性を評価する。また、複数のエージェントが同時に移動する場合に、効率良く不要

なコード転送をキャンセルするアルゴリズムを検討する。

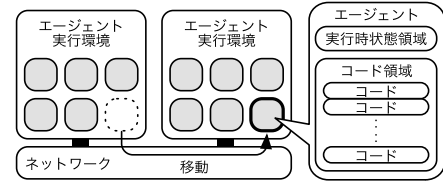


図1 モバイルエージェントの概要

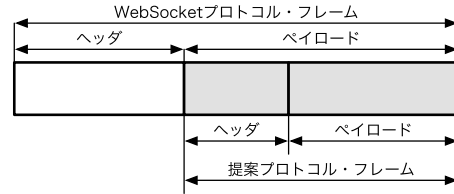


図2 提案プロトコルのフレーム構造

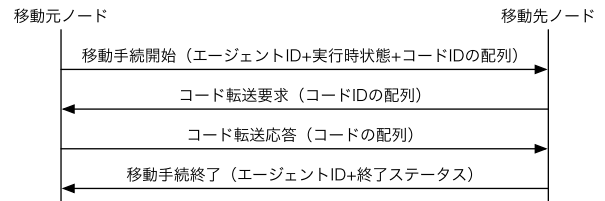


図3 コード転送が必要な場合のシーケンス図

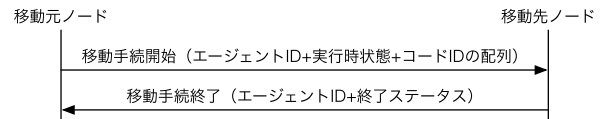


図4 コード転送が不要な場合のシーケンス図

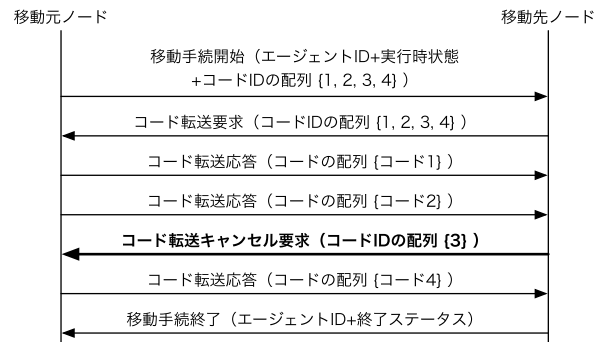


図5 コード転送キャンセル要求が発生した場合のシーケンス図

文 献

- (1) Russell, A., "Comet: Low Latency Data for the Browser," <http://infrequently.org/2006/03/comet-low-latency-data-for-the-browser/>, March 2006
- (2) Fette, I., "The WebSocket protocol draft-ietf-hybi-thewebsocketprotocol-10," <http://tools.ietf.org/html/draft-ietf-hybi-thewebsocketprotocol-10>, July 2011
- (3) Fuggetta, A., Picco, G.P. and Vigna, G., "Understanding Code Mobility," IEEE Transactions on Software Engineering, Vol.24, pp.342-361, May 1998