

Java を用いたモバイルエージェントによる障害推測

2H-3

金子真人[†], 大喜多優[†], 望月祐洋[†], 徳田英幸[†][†]慶應義塾大学大学院 政策・メディア研究科[†]慶應義塾大学 環境情報学部

1 はじめに

本研究では、モバイルエージェントがネットワーク上のホストの故障診断 [1] を行い、故障が存在する場合は故障原因の推測を行うシステムの設計・実装を行う。

モバイルエージェントを利用する理由として、以下に示されるような特徴が挙げられる。[2]

モバイルエージェントには、相手先のホストに実行可能なプログラムを輸送して処理を行う「マイグレーション機能」があるため、相手先のホストと常に接続する必要がなくなり、ネットワークのトラフィック軽減が期待される。また、接続が不安定なモバイルコンピューティング環境下でも継続的な処理を実現できる。

また、Java の使用により、プラットフォームや OS に依存しないシームレスなコンピューティング環境を実現可能である。

本稿では、モバイルエージェントを用いることにより、「接続が不安定な環境下での利用」「シームレスなコンピューティング環境」という特徴を活かした故障診断システムの実現に関して述べる。

2 モバイルエージェントによる故障診断

モバイルエージェントを利用したシステムの一例として、PDA のような携帯情報端末からネットワーク上の、セグメントの下のホストの故障を診断することを想定したシステム、「Agent-De-Go(以下、ADG)」を考案した。本システムは、ポートが落ちている、ファイルがマウントされていない、物理的にホストがネットワークに接続されていない等の故障への対処を考慮しており、次のような機能を目指としている。

- 接続状況が不安定な環境を想定し、接続を切断しても診断を行うことができるようにする。
- 診断対象のホストに故障が発見された場合は、故障の種類と原因を具体的に示す。
- エージェントのプログラムで修復できる故障は、修復を行う。

3 ホストの接続状況の診断

本プロトタイプでは、前節の故障診断の具体例として、ホストの接続状況を診断する部分を実現した。

本システムは、診断するホストを指定し、そのホストにエージェントの送信を行い、診断後に故障原因の推測を行い、また推測後にクライアントにログファイルを返送するクライアントおよびサーバと、故障診断の対象となるホストで構成されている。

クライアント GUI に表示されるホスト名の一覧から診断したいホストとサービスを選択し、ADG サーバにそのデータを送信する。ADG サーバには複数の診断エージェントが用意されており、診断するサービスの種類により診断エージェントが選択される。エージェントは各ホストに移動して故障診断を行い、異常を発見する。発見された異常をもとに、原因を推測する。

この診断は、次のプロセスで行われる。

The conjecturing trouble for Mobile Agent using Java
Masato Kaneko, Yuu Okita, Masahiro Mochizuki and Hideyuki Tokuda

[†]Graduate School of Media and Governance, Keio University
5322, Endo, Fujisawa, Kanagawa 252-8520, Japan

[†]Faculty of Environmental Information, Keio University
E-mail:{gunchan,yuudra,moma,hxt}@sfc.wide.ad.jp

1. データを ADG サーバへ送信

ADG クライアントで、通信を行う ADG サーバ名と、診断を行うホストの一覧・サービスを指定し、ADG サーバに送信する。

2. ADG サーバによる診断エージェントの送信

ADG サーバは、指定された、診断したいサービスの種類を元に診断エージェントを選択し、クライアントで指定した診断対象のホストに診断エージェントを送る。

3. 診断エージェントによるホストの巡回

選択された診断エージェントは、指定されたホストを巡回して接続状況の診断を行う。ホストが物理的に接続されていない場合は、ホストに到達できないので、ADG サーバにエラーメッセージを送る。

4. 診断サーバによる故障診断

診断対象のホストでは、診断サーバが起動されている。診断サーバは、エージェントから情報をコピーして故障診断を行い、診断結果を診断エージェントに返送する。エージェントの種類に応じた故障診断が実行可能である。

5. 診断エージェントによる診断結果の通知

診断エージェントは、1つのホストの診断が終了すると、ADG サーバにホスト名と診断結果を送信する。ADG サーバでは、これを診断ログとしてファイルに保存する。保存した後、診断エージェントは次のホストの診断を行う。すべてのホストの診断が終了すると、診断エージェントはサーバに戻り、診断完了通知をする。

6. 故障原因の解析

ADG サーバでは、診断ログをもとに障害の種類を解析し、原因を推測する。推測結果は推測ログとしてファイルに保存する。

7. 診断結果の受信

本システムでは、接続が不安定な環境下での利用を想定しているため、推測ログは直接クライアントに返送せず、一旦 ADG サーバで保存する形式を採用している。また、故障診断・推測と、その結果の受信は別々の処理で実行できる方式とした。

診断結果の受信は、故障診断処理とは独立して実行する。すべての故障において推測が完了した後、ADG クライアントから ADG サーバに接続すると、最初に ADG サーバから ADG クライアントに全診断終了通知が送られる。その後、診断ログと推測ログを受信する。本システムでは、クライアントは、最初に診断情報を ADG サーバに送信する時と結果を受信する時だけ ADG サーバと通信を行えば良い。

4 設計

本システム概念図を図 1 に示す。

本システムで診断対象とするホストは、Java をインストールしているものとする。

まず、本システムを情報収集部と原因診断部の 2 つに大別する。

診断結果の受信の部分は、ADG クライアントおよび ADG サーバから独立して Receive クライアントおよび Receive サーバとする。

情報収集部と原因診断部での ADG クライアント、ADG サーバ、診断サーバ、Receive クライアント、Receive サーバ、診断エージェントの役割を示す。

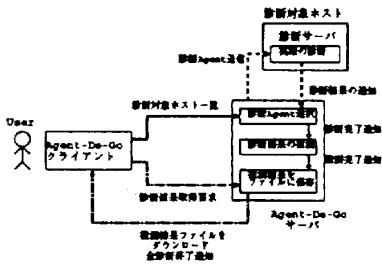


図 1: Agent-De-Go 概念図

4.1 情報収集部

情報収集部は、ADG クライアントおよび ADG サーバ、Receive クライアントおよび Receive サーバ、診断サーバ、診断エージェントで構成される。以下にそれぞれの役割を示す。

- ADG クライアント

診断を行うホストを選択し、ADG サーバに送信する。
- Receive クライアント、Receive サーバ

ADG サーバで故障診断と推測が終了した後、Receive クライアントから Receive サーバに接続し、診断ログと推測ログをダウンロードする。
- ADG サーバ

ADG クライアントから指定されたサービスをもとに診断エージェントを選択し、クライアントから指定されたホストに診断エージェントを送信し、1つのホストの診断が完了するごとに診断結果を診断ログとしてファイルに保存する。指定されたホストすべての診断が完了するまでこの作業を行う。
- 診断エージェント

ADG クライアントから指定されたホストの巡回を行う。指定されたホストの診断サーバに診断するサービスの情報を渡し、診断サーバから診断結果を返送されると、その結果を ADG サーバに診断ログとして返送する。診断サーバに接続できない場合は、接続は行わず、エラーメッセージを診断ログとして返送し、ADG サーバに戻る。
- 診断サーバ

診断対象ホストでは、診断サーバを実行し、診断エージェントからの接続を待つ。接続されたら、診断エージェントを受信し、情報をコピーし、実際の故障診断を行う。診断の種類は、診断エージェントの種類によって変わる。診断結果は診断エージェントに返送する。

4.2 原因診断部

原因診断部の作業を行うのは、ADG サーバのみである。ADG サーバは、診断エージェントによるホストの診断がすべて完了して、診断完了通知を受信したら、診断ログもとに、故障原因の推測を行う。ログのエラーメッセージをもとにして推測を行い、それを推測ログとしてファイルに保存する。

5 実装

本プロトタイプは、前節の設計のうち、情報収集部を実装した。

ADG クライアントに Let's note AL-N1T515J5 (Pentium150 MHz, Windows95), ADG サーバに Sun Enterprise3000 (Solaris2.5.1) を使用しており、実装には JDK1.1.5 と HORB1.3b1, GUI には swing-1.0.2 を用いた。クライアントとサーバ間の通信は HORB[3] で実現している。診断サーバは ServerSocket で実装しており、エージェントと診断サーバ間の通信は Socket で実現している。

情報収集部の動作の流れは次のようになる。

1. まず、ADG サーバで HORB を起動し、診断対象のホストで診断サーバを起動する。

2. メニュー画面は、Java アプレットで実装しており、ADG クライアントと Receive クライアントを起動できる。ADG クライアントでは、診断したいホスト名をリストから選択する。また、通信する HORB サーバ名とそのポート番号、出力ファイル、診断するサービスを入力する。ホスト名は、ADG クライアント上で追加が可能である。
3. 入力した情報を ADG サーバに送信する。
4. ADG サーバでは、指定されたサービスの種類から診断エージェントが選択される。診断エージェントは、指定されたホストの診断サーバにソケットで接続を行う。ホストが物理的に接続されていないは、接続は行われず、例外処理で ADG サーバに診断ログとしてエラーメッセージを送る。
5. 診断サーバは、診断エージェントのインスタンスを作り、メソッドを実行することにより、実際の故障診断を行う。診断エージェントの種類に基づいた診断を行うことが可能である。診断結果はストリームで診断エージェントに返送する。
6. 診断エージェントは、指定されたホストの診断が完了すると、診断完了通知を行い、同時に、ADG サーバは診断ログをファイルとして保存する。
7. 診断完了後、Receive クライアントは、Receive サーバと接続を行い、診断ログをクライアントにダウンロードする。すると、全診断終了通知が出され、故障診断処理が終了する。

6 まとめと今後の課題

本プロトタイプでは、ADG サーバと Receive サーバの2回に分けて ADG サーバへの接続を実行することにより、クライアントとサーバを常に接続する必要がなくなる。また、Java で実装したことにより、診断対象のホストのプラットフォームや OS に依存しない故障診断が実現可能となる。診断サーバは、診断エージェントを受信してコピーする方式を採用しているため、診断エージェントはマイグレーションを実現している。

しかし、現在の実装では、Java の例外処理を用いて障害の区別をしているので、故障の種類や原因を具体的に示しているとはいえない。

今後は、原因診断部の実装も進める予定である。

本システムでのこれからの課題を挙げる。

- エージェント自身が故障を起こした場合

診断エージェントが作業中に故障を起こし、異常な出力を行うようになった場合に対し、そのエージェントを破棄して新しいエージェントの送信を行う等の対策を考える。
- 診断時間の短縮

診断するホストがネットワークに接続されておらず、接続タイムアウトになる場合などは診断に時間がかかってしまう。タイムアウトになる時間を指定する、経路制御も調べてネットワーク中のどの部分で故障が発生しているか調査する等の対策により診断時間の短縮を検討する。
- ホストの並列診断

診断するホストが多い場合も診断に時間がかかってしまう。そのため、多数のエージェントで同時並行的に診断を行うことで、診断時間の短縮を図る。

参考文献

- [1] 千村文彦, 所真理雄:「理論修正と故障診断の統合について」, 日本ソフトウェア科学会 第2回 マルチエージェントと協調計算ワークショップ, 1992
- [2] Danny B.Lange, Mitsuru Oshima, Kazuya Kosaka: 「Aglets: Programming Mobile Agents in Java」, WW-CA97, March 1997
- [3] Satoshi Hirano, 「HORB: Distributed Execution of Java Programs」, WWCA97, March 1997