

## 信頼性の高い移動エージェントシステムの構成方法\*

5U-7

新堀健治†

渡部 卓雄†

北陸先端科学技術大学院大学 情報科学研究科

## 1 はじめに

本研究の目的は、信頼性の高い移動エージェントシステムの機構を考案し、その実装を試みることである。既存の移動エージェントシステムである Aglets [1] などは、計算機環境に障害に関して十分考慮された設計になってない。そのため計算機環境に障害が発生した場合、処理を完結することができない。本研究が対象とする移動エージェントは、1) レプリカによる矛盾が起きない仕事だけをサポート、2) 各ホストで行う仕事はどのホストでも実行することができる、ことを前提とする。この時、計算機環境で発生する様々な障害を分析し、それぞれの障害レベルに応じた処理を例外処理として実現する機構を提案する。

## 2 障害

現在、下記のような計算機環境の障害があると考えている。

1. ホストがダウンした場合。
2. 通信が不可能になった場合。
3. エージェントが時間内に帰ってこなかった場合。
4. 移動先で期待していた資源が使えなかった場合。

障害1、2、3はいつ障害が回復するかわからないホスト又はエージェントを待つよりは、別のホストに移動して計算を続けた方が良いと思われる。この3つの障害に関してはタイムアウトを目安に障害検知をする予定である。障害4に関しては資源を予め明確にしておかなければならない。具体的な例としては、1) 待ち合わせていた移動エージェントがいなかった、2) ファイル等の情報が手に入らなかった、

3) メモリやCPUの個数など、期待していた性能が得られなかった、などが考えられる。移動エージェントの仕事が障害が発生したホスト以外でも実行可能なら、別のホストへ移動して仕事を続けるか、障害ホストで仕事をしたい場合は障害が回復するまで待つことになる。ただし、各ホスト全体がダウンしてしまうようなシステム全体に致命的な障害が起きたときは、本研究の範囲を超えているので、そのような障害に関しては今回は対象としない。

## 3 エージェントサーバー

各移動エージェントサーバーは、図1のようなTree状に接続する。この特徴として移動エージェントは行き先のアドレスを知らなくても巡回できることと、各ホストを種類別のグループに分けることもできる。図1からわかるように、Treeの上から下へ行く場合は、幾つかホストへ行く路があるが、下から上へ移動するような場合は経路が限られてしまう。そこで監視ホスト間を結ぶことで、移動エージェントの移動経路の選択の幅を増やしている。ただし巡回するときは、ループしないように規則を定める必要がある。移動エージェントのプロトコルは、プロトコルエージェントによって管理されている。Treeの構成としてはRootに近いマシンほど信頼性の高いマシンになるようにして、ノートパソコンなどのネットワークに常時接続されていないマシンは葉になるように構成する。Agent Dataを備えた監視ホストはネットワークに常時接続されていて、信頼性の高いマシンを選ぶ。

## 3.1 Agent Data

主にエージェントのレプリカを保存するためのものである。このレプリカはエージェントの目的が完全に終わったと確認されるまでは、各管理ホストにレプリカは保管されている。エージェントの仕事が完全に終了した時に、監視エージェントによってレプリカは

\*A Framework for Reliable Mobile Agents

†Kenji SHINBORI, Takuo WATANABE

{ken2, takuo} @jaist. ac. jp

Graduate School of Information Science, Japan Advanced Institute of Science and Technology

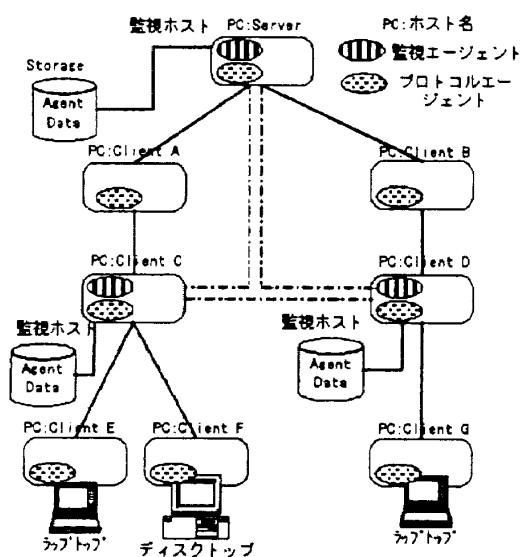


図 1: サーバーの構築例

破棄される。

### 3.2 監視ホスト

監視エージェントと Agent Data を備えたホストのことで、監視ホストで移動エージェントのレプリカなどの管理を行う。監視エージェントは、マシンの監視を行い、タイムアウト等の情報をシステムに知らせたり、レプリカの削除を行う。監視ホストで監視をするマシンは、監視ホスト間のマシンと、子が葉となるホストがある場合はそのホストまでを監視する。監視ホストの位置は、ホストの子が複数存在するときにそのホストを監視ホストにするべきだと考えている。監視ホストの数を増やすことは、信頼は向上するが逆に冗長も増すことになり、適切な構築方法を考えなければならない。

## 4 障害対処機構

本研究の移動エージェントシステムの障害処理機構は障害に対するイベントと処理部分（メソッド）からなる。障害が発生した時、移動エージェントはシステムから障害イベントを取得し、それに対応する処理を実行する。この処理部分を記述には Dependability Installation Language (DIL) [2] の概念を用いる。この DIL は、監視方法や障害処理（プロトコル）を記述するものである。DIL には、幾つかの基本メソッドがあり、これにより汎用的なプロトコルが定義される。また、メソッドをオーバーライドすることができるこ

とと、幾つかの基本メソッドをインクリメンタルに追加することによって、プロトコルをカスタマイズすることができる。この機構の利点としては、

- 最低限の障害処理は基本メソッドで保証されている。
- 個々の移動エージェント特有の障害処理を実現するには、基本メソッドのオーバーライドやメソッドの追加で実現することができる。

プロトコルはプロトコルエージェントが管理している。全体で共通なプロトコルと、個別のプロトコルがあり、各移動エージェントは対応したプロトコルでのみ動作することを保証する。プロトコルエージェントを使うことにより、プロトコルを動的に作成、変更できる利点がある。

## 5 実装及び今後の課題

実装は Java ベースのエージェントシステムである AgentSpace [3] を拡張する予定である。今後の課題は、1) 基本メソッドやエージェントの管理方法などを明確にすること、2) 実装後、監視ホストの位置や数を実験により最適と思われる構築方法を導く、3) リフレクションを用いることで、システムの変化に対して柔軟に対応するプロトコルの機構を考案する、以上が今後の課題がある。特に3) はすでに動作しているエージェントのプロトコルを動的に変更することは、容易にはできないと思われるので、その機構に関しては十分考察する必要があると思われる。

## 参考文献

- [1] Danny B. L, et al.: Aglets Programming Mobile Agents in Java, LNCS 1274, pages 253-266, Springer-Verlag, 1997.
- [2] Daniel C. S and Gul A. A.: A Protocol Description Language for Customizing Failure Semantics, Proceedings of 13th International Symposium on Reliable Distributed Systems, pages 148-157, 1994.
- [3] 佐藤一郎.: AgentSpace 高階モバイルエージェントシステム, 電子情報通信学会技術研究報告, Vol97, No627, pages 41-48, 電子情報通信学会, 1998.