2ZB-3

移動エージェントを用いたフォルトトレラントシステムのデータ応答手法

山口 俊一† 森 秀樹† 上原 稔†

東洋大学工学部情報工学科†

1. はじめに

コンピュータ技術と通信技術の進歩により、ネットワークを介した分散処理による、資源の効率的な利用が可能になった。これらの多くは常時稼動している状態にあり、また、重要なシステムとして利用されている。この時に運用上考慮しなければならないのがシステムの障害や故障、誤りに対する耐故障性（フォルトトレラント）である。これを回避する方法として、一般的にはシステムに専用のハードウエアを多重に構築し物理的な故障を回避する方法や常時バックアップを行う方法、チェックポイント、スナップショット、TMR、などの分散アルゴリズムによる方法、分散 OS、負荷分散機構(LSF)[1]などを複合的に導入することによりフォルトトレラントを実現している。しかし、これらの多くはそのシステム専用に用いられているため汎用性やスケーラビリティ（規模適応性）といった点で問題がある。そこで、補完技術として移動エージェント技術が注目されている[2]。

移動エージェントシステムはエージェントがネットワーク上を自律的に移動することができ、実行状態を保持し完全に移動した後に実行を再開する。サーバ側の処理が終了するまでネットワークトラフィックが発しないので、RPC に比べ通信回数を最小限に押さえることができ、ネットワークにかかる負荷を少なくすることができ、故障箇所を容易に回避させることも可能となる[3]。そこで我々は、移動エージェントを用い演算結果を多数決処理する耐故障性を考慮したシステムを提案した[4]。IBM 基礎研究所で開発された Java による移動エージェント実行環境 Aglets[5]を用い、実行される移動エージェントを複数用意し、異なるノードでそれぞれ実行させ、ユーザータスクの実行結果を多数決することにより実行結果に対する耐故障性を持つシステムの構築を行った。しかし、ノード上にあるエージェントの行った実行結果を任意のコンピュータから確実に取得する部分が課題となっていた。そこで、本論文では、提案システムでエージェントがネットワーク上のノードで計算した最終的な計算結果を、耐故障性を確保した状態で任意のコンピュータから取得する手法を議論する。

The data response method of Fault Tolerant system using Mobile Agents.

†Shunichi Yamaguchi, †Hideki Mori, †Minoru Uehara,
Department of Information and Computer Science, Toyo University

2. システムのアーキテクチャ

各ノードは Java バーチャルマシン（JVM）を持ち、その上で移動エージェントが実行され、ユーザータスクが処理される事になる。タスクは到着ノード、またはネットワークを介した移動先のノードで処理される。後者の場合、その処理結果はネットワークを介して元のユーザーへ返送されることになる。ユーザーからはタスクの移送が透過的に見え、システム全体を一つのコンピュータであるように利用することができる。具体的なシステムの構成要素としては以下の通り。

・Master Agent
各ノードに常駐するコンポーネントで、ユーザーとのインターフェースの表示、ユーザータスクの受付、タスク ID の決定、処理ノードの決定、Slave Agent の生成、Shared Agent へのタスク ID・Slave Agent 位置情報の登録、Slave Agent により送られてきた最終的な処理結果の多数決、結果の表示を行う。

・Slave Agent
Master Agent によって生成されるコンポーネントで、ユーザータスクと処理結果を所持しながら、タスク投入ノードと処理ノードの間を移動してユーザータスクを実行し。結果を多数決処理し最終結果を保持し、Master Agent へ送信する。

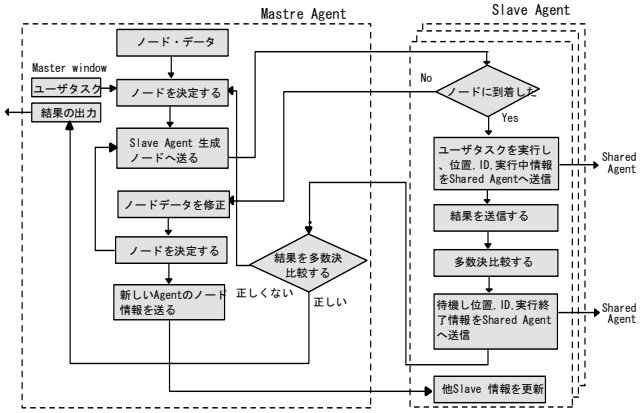

図 1 移動エージェントの基本動作

・ユーザータスク
ユーザーの実行すべき Java のクラスファイル。

Java アプリケーション。

・Shared Agent

アドレスを元に利用可能ノードを機械的に 3 つ以上のグループに分け、それぞれのグループの特定のノードに 1 つの Shared Agent が常駐する。

Slave Agent の移動先情報と保持しているタスク ID、状態の 3 つの情報を持つ。

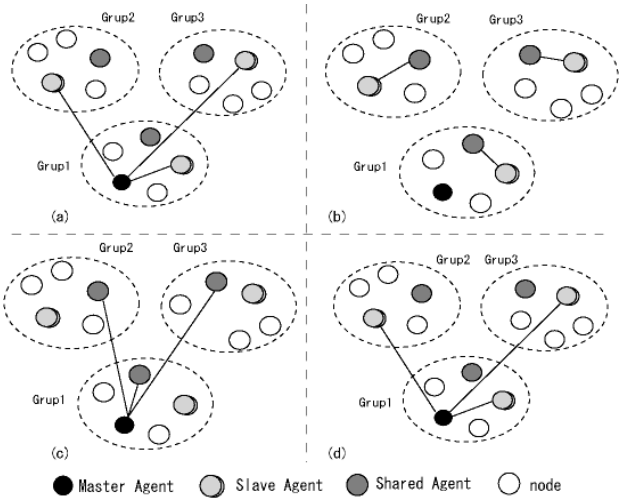

図 2 各 Agent の応答動作

3. 実装システム

3.1 移動ノードの決定

生成された Slave Agent の移動先である処理ノードの決定は Shared Agent の情報、Slave Agent の移動により移動不成功という通信のあったノード情報と多数決処理によって途中通信の無かったノード情報を元に Master Agent が行う。

3.2 多数決処理

生成された Slave Agent は移動先のノードでユーザータスクを実行する。本システムでは、同じユーザータスクを持つ Slave Agent を 3 つ生成し、それぞれ異なるノードで実行させ、最終的に実行結果を多数決処理することにより、実行結果については N-1 の故障について耐故障性を実現している。

実行が処理終了した Slave Agent は現在いるグループの Shared Agent に位置情報とタスク ID を登録する。

4. 実行結果データの取得手法

各 Slave Agent で実行された実行結果は従来では、タスクを投入したノードの Master Agent を利用するしかなかった。しかし、この投入ノードが故障した場合や、待機している Slave Agent が応答しなくなった場合、実行結果を取得することができなくなってしまう。そこで、Shared Agent を用い、任意のノードから結果を取得する方法を述べる。

まず Slave Agent で実行された結果を取得するために、Master Agent は実行時に Shared Agent を探し、見つかった Shared Agent に対してタスク投入時に設定したタスク ID と Master Agent の位置を送信する[図 2(c)]。Shared Agent は受け取った情報を元に、自分の持つ Slave Agent の位置情報とタスク ID を比較し、該当するものがあった場合には状態情報を参照し、実行中であれば、Master Agent に実行中のメッセジーを送信し、待機中であれば、Master Agent に実行結果を送信する命令を該当する Slave Agent に出し、Master Agent は結果を受信する[図 2(d)]。

Shared Agent を用いてノードをグループとして管理し、タスクを持つ Slave Agent の位置を保持することによって、利用者はタスク ID のみ記憶しておくことにより、実行結果を受け取る端末を特定することなく、利用することができる。また、複数のグループに同一のタスクを実行した Slave Agent があるため、それらの結果を比較することにより、信頼性が向上が期待でき、Slave Agent へのデータ送信要求もブロードキャストを行うよりも少なくてすむ。

5. まとめ

本論文では、分散環境のフォルトトレラント手法として移動エージェントを用いたシステムの演算結果データを接続している任意の端末から取得する手法について述べた。また、議論によって、本システムにおいて任意の端末から実行されていた演算の結果を、透過的に取得する手法として有効であると言える。今後、本手法を実装し評価実験を行っていくと共に、動的に構成の変化する分散ネットワーク環境やロードバランシングに対応したシステムの実装を行っていく。

参考文献

[1] Andrew S. Tanenbaum and R. Van Renesse, “Distributed operating systems,” ACM Computing Surveys, vol.17.no.4, pp429-470, Dec.1985

[2] 本位田真一, 飯島 正, 大須賀昭彦, “エージェント技術”, 共立出版, 1999,

[3] Andrew S. Tanenbaum, Maarten van Steen, “分散システム 原理とパラダイム”, ピアソン・エデュケーション,

[4] 山口 俊一, 森 秀樹, 上原 稔,”移動エージェントによる多数決を用いたフォルトトレラントの研究”, 信学技報 FIIS04, No142, pp.1-7 (2004.6)

[5] IBM 基礎研究所, Aglets. http://www.research.ibm.com/trl/aglets/