6 E − 1

Java による分散共有空間上での移動エージェントの実現

山本 勉　鈴木 寿郎
株式会社沖テクノシステムズラボラトリ

1 はじめに

Linda[1] に強く影響を受けた分散共有空間 JavaSpace[2] が, Java[3] に対し Sun から提案されている。しかし, この空間で送受されるのは基本的にオブジェクトのデータだけである。

プログラム・コードの非同期的な送受を含む動的な分散処理を可能にするため, 我々はこの空間上に移動エージェントを構築した。ここでは, このエージェントのシステムの概要, 実装方法, 応用例を述べる。

2 移動エージェントシステムの概要

本システムは 図1 に示すように JavaSpace, 移動エージェント, エージェントサーバから構成される。エージェントサーバはネットワークにまたがって存在できる。移動エージェントは, 共有空間である JavaSpace を経由してエージェントサーバからエージェントサーバへと移動するとともに, それぞれのエージェントサーバ上で自分の果たすべき処理を遂行する。

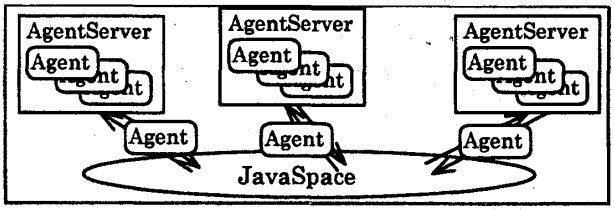

図1 移動エージェントシステム

2.1 JavaSpace

本システムはエージェントの移動経路および対エージェント用の非同期的な通信経路として JavaSpace を使用する。本実装で用いた JavaSpace は, Sun が [2]で発表した仕様にもとづいて我々が独自に作成したものである。

JavaSpace は, Linda のタプル空間に相当する分散共有空間である。ネットワークにまたがって散在する各 Java プログラムは, JavaSpace に対し図2のようにクライアントとして以下の操作をすることができる。

・エントリの書き込み (write)
・エントリの読み出し (read) と取り出し (take)
・write イベントの通知の受信 (notify)

エントリは, Linda のタプルに相当するオブジェクトであり, 利用者が任意にクラス派生で定義し構築する。read,take,notify の各操作において, エントリは型の適合性と非 null フィールド値の等価性にもとづくマッチングによってこの空間から選択される。

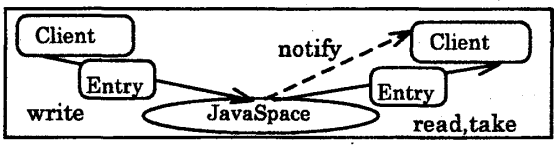

図2 JavaSpace システム

2.2 移動エージェント

移動エージェントは, 自分自身の状態を保持しつつエージェントサーバ間を移動できる Java オブジェクトである。エージェントは移動時に自分自身の状態と自分自身を構成するクラスのバイトコードを JavaSpace を介して移動先へ転送する。これにより, 移動先にオブジェクトの状態と振舞を送り込む。移動先に移動エージェント固有のクラスファイルをあらかじめ用意しておく必要はない。

プログラム例を図3に示す。利用者は, 移動エージェントを, 基底クラス Agent の派生クラスとして任意に定義する。

メソッド Agent#toSpace(JavaSpace space, String where) を呼び出すと, 移動エージェントが, space 経由で, 名前 where を持つエージェントサーバへと転送される。移動先のエージェントサーバに到着したとき, メソッド Agent#act(where, space, aux) が呼び出される。aux はエージェントサーバが任意に与える付加的な引数である。移動エージェントの振舞はその act メソッドによって定義される。

```
import jive.agent.*;
import jive.javaSpace.*;
import jive.util.*;

public class MyAgent extends Agent {
    protected String message = "Hello Agent";
    // コンストラクタ
    public MyAgent() {}
    public MyAgent(String name) { super(name, ""); }
    // 移動先(エージェントサーバ)でコールバックされるメソッド
    public void act(String where, JavaSpace space, Object aux) {
        System.out.println("Here is " + where + ".");
        System.out.println("Message: " + message);
        /* ここで toSpace("somewhere", space)を呼ぶと somewhere
           という名前のエージェントサーバへ移動する */
    }
    public static void main(String[] args) {
        try {
            JavaSpace space =
                JavaSpaceService.getSpace();        //JavaSpaces へ接続
            MyAgent agent = new MyAgent(args[0]);//エージェントの生成
            agent.toSpace(space, args[1]);          //エージェントの移動
        } catch (Exception e) {
            e.printStackTrace();
            System.exit(1);
        }
    }
}
```

図3 エージェントのプログラム例

An implementation of mobile agents on the Java tuple-space systems
Tsutomu Yamamoto and Hisao Suzuki
Oki Technosystems Laboratory, Inc.

2.3 エージェントサーバ

エージェントサーバは，移動エージェントが活動する場所を提供する。エージェントサーバの実体は，AgentServer クラスのオブジェクトを持った，JavaSpace クライアントである。

エージェントサーバは自分の名前を持っており，自分宛ての移動エージェントがJavaSpace へ書き込まれると，その移動エージェントをJavaSpace から取り出し，そのエージェントが定義している Agent#act メソッドを呼び出す。

3 実装

本システムはJDK 1.1.x があれば動作する。現在，Windows95, Linux 2.0, Solaris 2.5 での動作を確認している。

本システムの実装の要点は，クラス定義（これにはプログラム・コードも含まれる）を含めたエージェントの生成，移動，復元にある。以下，その過程を説明する（図4)。

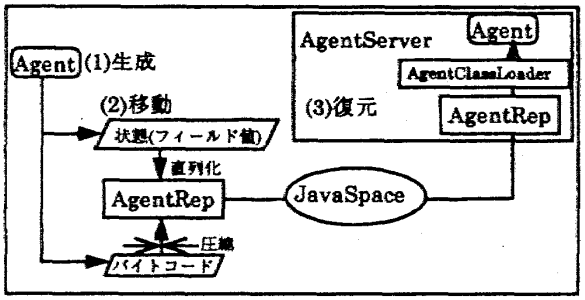

図4 エージェントの移動過程

3.1 エージェントの生成

移動エージェントは Agent の派生クラスのオブジェクトであり，他のオブジェクトと同様，単に new 演算によって生成する。

3.2 エージェントの移動

エージェント移動メソッド Agent#toSpace は，エージェントに対し，まず Object Serialization API[3]を使用して各フィールド値を直列化するとともに，もしまだ未取得ならば Reflection API[3]を使用して自分自身の再構築に必要なクラス定義のバイトコード群を取得する。そしてそれらをエージェントの移動時の表現型である AgentRep オブジェクトの中へ収め，それを JavaSpace へ書き込む。AgentRep は JavaSpace のエントリとして定義されている。

エージェントサーバは自分宛てのAgentRep をJavaSpace から選択的に取り出そうと待機している。JavaSpace に書き込まれた AgentRep は，このようなエージェントサーバに非同期的に拾われる。

3.3 エージェントの復元

エージェントサーバは，拾った AgentRep が収めていたバイトコードと直列化フィールド値から，エージェントオブジェクトを復元する。そして，その act メソッドを呼び出すことによってエージェントの活動を再開させる。復元時には，専用のクラスローダ AgentClassLoader がAgentRep からのバイトコードのローディングに使われる。本 JavaSpace の実装は RMI[3] を使用しているが，RMIClassLoader のような標準的なローダは，同名かつ異内容のクラスを同時に並行してロードし得るような環境，復元時にエージェントの生成元のサーバがアクセス可能と必ずしも保証されていない環境では利用できない。

4 応用例

移動エージェントの応用として，特定の種類のクライアントを巡回して情報を収集するアプリケーションを開発した(図5)。エージェントは，各クライアントに AgentServer オブジェクトとして組み込まれたエージェントサーバを巡回する。そして Agent#act の aux 引数に対するメソッド呼出しによって，宿主であるクライアントの情報を収集する。

現在，その種のクライアントとして，インターネット上の情報を加工して再配信するもの，ローカルファイルシステム上のファイルの内容を情報として配信するもの等を用意している。

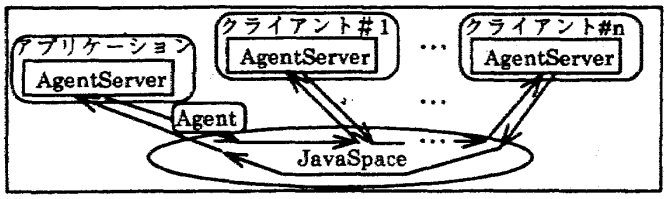

図5 移動エージェントの応用例

5 おわりに

本システムのエージェントの移動は，JavaSpace という分散共有空間上の AgentRep エントリの非同期的な授受によっている。これは，エージェント移動の際，移動先や生成元など，それに関係するエージェントサーバが終始すべて作動状態になくてもよいことを意味する。

また，エージェントの移動先は，JavaSpace 上のエントリ AgentRep のエージェントサーバ名を表すフィールド値のマッチングによって決定されているだけである。同名のエージェントサーバを複数設けることにより，Linda の eval タプルに相当する並列的で動的な分散化計算を実現できる。

現在，本エージェントとスクリプト言語を組み合わせて，より柔軟な分散計算を行う試みが別途計画されている。

謝辞

本研究の機会を与えてくださいました沖電気工業株式会社研究開発本部メディアネットワーク研究所の中澤修プロジェクトオーガナイザーに深く感謝致します。

参考文献

[1] Nicholas Carriero and David Gelernter. *How to Write Parallel Programs: A First Course*, MIT Press, 1990.

[2] JavaSpace Specification Revision 0.4, http://chatsubo.javasoft.com/javaspaces/js-spec/js.pdf

[3] Java Development Kit 1.1.4 Documentation.