

分散環境における金融意思決定支援システムの設計

2N-4

溝口文雄* 長谷川誠*

東京理科大学 理工学部†

1 はじめに

近年インターネット／イントラネットの急速な普及で分散システムの需要は高まりつつある。グローバル化する金融の分野に関しても同様で、フロー情報のリアルタイム処理や分散型データベースといった分散システムの構築が必要とされる。ソケット通信を用いた従来のサーバ・クライアントモデル（階層モデル）は、複雑化する分散環境に柔軟な対応をすることが困難である。しかしネットワーク上の分散システム構築のために作成されたJava RMIを用いることによって、分散システムの構築手法は大きく変貌し、階層的なモデルから散層的なモデルへと移行することができる。そこで本研究では、以前作成された金融意思決定支援システム [1] を基にJava RMIを用いて [2]、散層的な分散システムの構築を行なう。

2 設計方針

金融システムを分散することの目的は、金融データ・ユーザのシステム操作の履歴といったデータを共有することであり、また複雑化するシステムを分散することで、拡張性が高くシステムの変更に柔軟に対応できるようにシステムを設計しなければならない。

本システムの設計方針をまとめると次の3点が挙げられる。

● データの共有

金融データのデータベース化やユーザ操作履歴ファイルを共有し、ユーザ名とパスワードによるユーザ管理することで、どの計算機環境を用いても、今までの操作のデータを利用することができる。

● リアルタイム性

金融取引などの即時性が要求されるシステムでは、クライアントに対しリアルタイムにデータを更新する必要がある。本システムではクライアントの要求に対してデータを提供するのではなく、データの更新をデータベースがクライアントへ配送する。

● 拡張性

システムを拡張または変更する際に、すべてのシステムを停止することは望ましくない。本研究では分散化されたシステムを動的に変更できるように設計を行なう。

3 ルーティング式分散モデル

従来のクライアント・サーバモデルは、サーバがすべてのクライアントの情報を把握する必要があったため、複雑なシステムへの対応が困難であった。そのため散層的な分散システムの構築を行ないシステムの処理を一箇所に集中することを避けねばならない。またクライアントとサーバが動的にバインディングを行なうことで必要がある。

ルーティング式分散モデルの構成は図1ようになる。

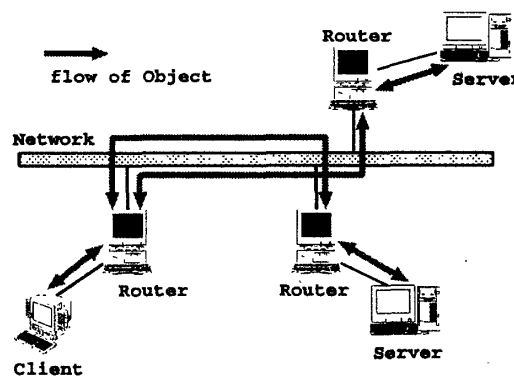


図1: ルーティング式分散モデル

ルーティング式分散モデルの特徴としては、サーバとクライアントは直接データのやりとりを行わず、必ずルータを介して行なうことである。ルータがシステムの動的にクライアントとサーバを結び付けることで、クライアントはサーバがネットワーク上のどのホストにあるのかを意識する必要がない。このモデルの長所はクライアントとサーバの結び付きが非常に薄く、お互いにインターフェースを介して処理を行なうため、ネットワーク上でどのように分散されるかを考えずに、それぞれ拡張することができる点である。

*Fumio MIZOGUCHI, Makoto HASEGAWA

†Faculty of Sci. and Tech. Science University of Tokyo

4 金融システムの構成

本システムでは、ルーティング式分散モデルを用いシステムの構築を行なう。システムの要素としては、サーバに金融データベース・ユーザファイルサーバを用意し、クライアントには金融クライアントがある。本システムではデータベース、ファイルサーバを通じてファイルへアクセスするためデータの共有がなされる。

4.1 金融データベース

オプションのデータをクライアントへ、日付とオプションの価格からなるデータオブジェクトをリアルタイムに供給する。データベースへの接続はJDBCを用いておこなう。次のコードはデータベースへの接続部分を示す。

```
public void update() {
    //データ取得のための SQL 文
    String query = "SELECT DATE,PRICE FROM FINANCE"
    //金融データベースからデータの取得
    ResultSet rs = stmt.executeQuery(query);

    Date today = rs.getDate(1);
    double price = rs.getDouble(2);
    PremiumTable c = getPremiumTable(today, price);
    //すべてのクライアントへデータを配送
    notifyObservers(table);
    ...
}
```

データベースから SQL 文でデータを取得し、notifyObservers メソッドを呼び出すことにリモートホストのクライアントへデータを送信する。物理的な理由でリモートホストに接続できない場合も考えられるが、そのときは、例外が発生しクライアントの接続が再び行なわれるまでデータの送信は行なわれない。

4.2 ユーザファイルサーバ

オプションの売買戦略の履歴、取り引き日等を含んだユーザデータをファイルへ保存する。データはネットワーク上で共有されクライアントはどこからでもこのファイルを読み込み書き込むことができる。保存するデータは OptionData としてまとめられ、その内容はオプションの戦略日付、価格等である。下記のコードはオプションデータのファイルへの書き込みを示す。

```
OptionData o =
    new OptionData("Buy_Call", "1/11", "345", ...);
fileserver.write(o);
```

4.3 金融クライアント

オプション売買を含むユーザ操作 GUI を持つ、GUI の構成としては、オプションプレミアムを表示するメインウィンドウ、ユーザのオプション戦略の履歴を表示するヒストリーウィンドウ、オプションの価格の変動をグラフで表すプライスウィンドウなどで構成される。

5 リアルタイムデータの処理

金融データをリアルタイムにデータベースから提供するための仕組みとして、デザインパターンの1つである Observer パターンを分散環境に拡張した RemoteObserver パターンを用いている。Observer パターンは、オブジェクトの1対多の依存関係を定義するためのデザインパターンであり、あるオブジェクトが状態を変えたときに、それに依存するオブジェクトに自動的に知らせ更新される。RemoteObserver パターンの構造を図2に示す。

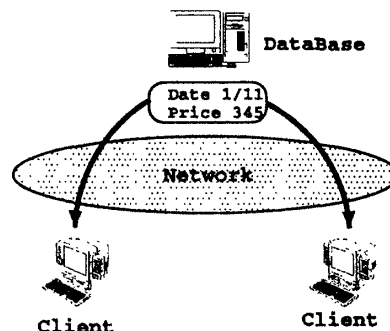


図2: RemoteObserver パターンの概念図

実装は RMI を用いることで容易に実装できる。このパターンを用いることで、オプションデータのようなフロー情報をリアルタイムに扱うことが可能になる。本システムではデータベースサーバが、日付とオプション価格のデータをあらかじめ登録されたクライアントへリアルタイムに供給する。

本研究室の実験環境においては、10 クライアントに対してオプションデータを配送するまでの時間は、クライアントでのデータ処理に関わらず、平均 2000ms で送ることが可能である。

6 おわりに

本稿では、Java のクラスライブラリ RMI を用いて、分散システムの構築を行なった。その手法としてルーティング式分散システムを用いた。またリアルタイムにデータの更新を行なうために RemoteObserver パターンを用いた。これによりデータの更新と同時にデータを転送を容易に実現できる。

参考文献

- [1] 溝口文雄, 大和田勇人, 木下信幸, 長谷川誠, Java による金融意思決定支援システム (その1) - システム設計 -, 情報処理学会第 53 回全国大会 (1996)
- [2] Jim Waldo, Geoff Wyant, Ann Wollrath, Sam Kendall, A Note on Distributed Computing (1994)