

## スケーラビリティと耐故障性を備えたインターネットの インテリジェントサービスの構築法<sup>1</sup>

1 J-3

水野伸太郎, 小野諭  
NTTソフトウェア研究所

アルゴリズムを用いて行なうことで着呼側の位置情報の参照速度が早くなり、サーバの負荷も軽減されることを示した。次の図はデータリプリケーションの様子を表したものである。

### 1 はじめに

これまでインテリジェントなネットワークサービスは独自の通信網を持つキャリアしか提供することはできなかった。しかし、インターネットが全世界レベルにまで広がり、グローバルネットワークインフラとして利用できるまでに発展し、通信インフラを持たない事業者でもネットワークサービスを提供することが可能となった。また、移動体通信の発達により、複数の端末を移動しながら使い分けて利用するユーザーが増えてきたため、インターネット上でのインテリジェントなネットワークサービスの必要性が増してきた。それでは、インターネットでインテリジェントなネットワークサービスを提供するためにはどのようなシステムを構築すれば良いのか。

本稿では、インテリジェントなネットワークサービスを提供するのに必要な、高度なビジネスロジックを処理することができ、スケーラビリティ、耐故障性、セキュリティを考慮したシステムをインターネット上で構築する方法について考える。

### 2 インテリジェントなネットワークサービスの具体例と要件

通信キャリアは、カスタマを識別したインテリジェントサービスを提供する際、接続の信頼性と接続の早さを保つためにスケーラビリティと耐故障性、セキュリティを兼ね備えたシステムを構築している。これらの性質を持つためにはどのような技術が必要なのかを移動体通信の例をもとに考える。

移動体においてはカスタマが今どこセル内にいるかという情報を常に監視し、サービスを提供しなければならない。しかし、発呼側が遠くの相手を呼び出す際、相手の位置データが遠くのデータベースにしかない場合、データを参照するための通信時間のロスが出てしまう。また、データが一つのサーバにしかない場合、このサーバに何らかの理由でアクセスできなければサービスを提供することができない。そのため、データを冗長に分散し、かつ整合性を保つように制御を行なう必要がある。この際に問題となるのが位置データのアクセスの早さ、信頼性、耐故障性という要件とそれらの一貫性を維持するためのコストとのトレードオフである。この問題を解消しようとするのがデータのリプリケーションである。携帯電話網などで用いられているリプリケーション手法にHLR/VLRがあり、これは次のような方法でデータアクセス速度を高めている。

1. 発呼者のいるエリアにあるデータベースに相手情報を問い合わせる。
2. 相手の情報が見つけれない場合、相手のホームゾーンのデータベースに問い合わせる。

発呼者がいるエリアのデータベースに相手のデータのリプリケーションがされていれば、遠くのデータベースにアクセスする必要がなく、高速化がはかれる。しかし、移動体においては位置情報は頻繁に変更されるため、全てのデータベースを逐一更新し、整合性を保つには時間がかかってしまう。この問題に関して、文献[1]では、携帯端末ユーザーのプロファイル情報のリプリケーションを、ユーザ毎の行動範囲を考慮した独自の

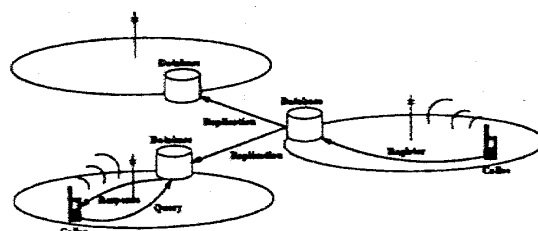


図1: 移動体通信におけるデータの分散管理

このように頻繁に変更され、かつアクセス時間が重要となるようなデータを扱う場合、サービスの信頼性を高めるためにはデータのリプリケーションが必要であることがわかる。

### 3 インターネット上でのインテリジェントサービス

移動体通信網や電話網は通信専用の網であり、インテリジェントサービスを行なうためにデータベースや処理系は専用に整備されている。また、セキュリティに関しても、ネットワーク自体が発呼者、着呼者の管理を行なっているために保証されている。

インターネットを見てみると、インターネットはグローバルでオープンな、コンピュータ通信のために構築されたネットワークである。そのために、ネットワーク自体には高度な機能はないが、ネットワークがオープンでスケーラブルであるという利点がある。そのため、独自のネットワークインフラを持たなくともインターネットを利用したネットワークサービスを提供することが可能である。

しかし、通信網のように統一整備された環境とは異なり、インターネット上には様々なOSで動く様々な計算機が接続されているため、マルチプラットフォームに対応する必要があり、ネットワーク自体にセキュリティがないため、認証などの機能も必要である。

また、前節でスケーラビリティと耐故障性を持つようなシステムを構築するには、ネットワーク上でデータを分散管理するための分散処理インフラが必要であることを述べた。

以上のことからインターネット上でスケーラビリティと耐故障性を持つ高度インテリジェントサービスを提供するために必要な要件として次のものが挙げられる。

1. 高度なビジネスロジックをネットワーク上で分散処理することができる
2. マルチマスター環境で利用できる
3. プラットフォームフリーである
4. カスタマ情報を分散管理するデータベース
5. セキュリティ

### 4 分散処理に必要なインフラ

前節で挙げた要件を満たすシステムを構築するには何が必要だろうか。RADIUSサーバの機能を拡張し、これらの要件

<sup>1</sup>Constructing intelligent services with scalability and fault tolerancy on the Internet  
Shintaro MIZUNO, Satoshi ONO  
NTT Software Labs.

を満たすシステムを構築する方法も検討されているが、我々はグローバルな分散環境での利用に重点をおいて考えていく。

まず、カスタマ情報をマルチマスタで分散管理するデータベース、セキュリティサービスは必要だが、これらと連動して高度なビジネスロジックを処理できるアプリケーションが必要である。本節では、これらの技術に関して、まずデータのリプリケーション技術、つぎに分散アプリケーションとそのセキュリティについて見てみる。

#### 4.1 データのリプリケーション

ネットワーク上の位置情報を管理する技術としては DNS サービスがあり、複数のマスターでデータを保持しているが、これはリードオンリーのデータベースであるため、高度なサービスの提供はできない。また、SQL データベースなどはデータを分散して保持でき、ストアドプロシジャなどを用いてある程度の処理を行なうことは可能だが、整合性を重視したシステムになっているので、頻繁に変更するデータのリプリケーションには向いていない。この要件に関連する技術として今注目を集めているのが、ディレクトリサービスである。ディレクトリサービスは、IETF の ASID WG など、検討されており、変更の多い情報を複数のマスタでキャッシュの一貫性を保ちながらディレクトリに蓄積するダイナミックディレクトリサービスが提案されている。これは通常のデータベースに比べ整合性よりもアクセス速度を重視したサービスとなっているため、位置情報等の管理技術として有効な技術である。このダイナミックディレクトリサービスには Microsoft 社が開発し、Cisco 社が UNIX にポートしている ActiveDirectory 等がある。

#### 4.2 分散アプリケーション

ディレクトリサービス技術については前に述べたように提案されているが、ディレクトリサービスはデータベースであるので、高度な分散処理を行なうことはできない。そのため、ディレクトリサービスと連動し、プラットフォームフリーで高度な分散処理を行なうことができる技術が必要である。

WWW 上のサーチエンジンもデータの位置を管理するサービスを兼ね備えているが、WWW は主にサーバからデータを読み込むという機能しか有しないため、高度インテリジェントサービスを行なうような処理を行なうことは困難である。最近ではプラグインやスクリプティング、ActiveX、Java アプレットなどで高度な処理を行なっているが、あくまでブラウザを介した処理であり、広くネットワーク上に存在するリソースと協調して処理を行なうということにはできない。

また、ここ数年で注目を集めている Java もインターネット上で高度な処理を行なうために利用されているが、Java は本来コードをネットワーク上に流すという形のクライアント/サーバタイプのシステムなので、ネットワーク上に点在するアプレットが協調処理を行なうということは困難であった。Java では Mobile Agent という技術を用いることにより分散処理を行なえるようにしようとしているが、まだ発展段階である。

それではこのようなシステムが構築可能な技術にはどのようなものがあるのかについて考えてみる。

##### 4.2.1 分散オブジェクト技術

グローバルなネットワーク上の分散処理インフラとして分散オブジェクト技術が注目を集めている。プログラミングの主流であるオブジェクトを用いた設計をすることができ、プログラミングもソケットや RPC などのように面倒な通信用プログラミングを行なう必要もなく容易であることで開発の負担も軽くなっている。

クライアントは下図のようにネットワーク上のいくつかのオブジェクトを利用し、分散処理を行なう。分散データを参照するクライアントがあり、クライアントはネットワーク上のオブジェクトをあたかも自分のマシン上にあるかのように利用することができる。これは分散処理を行なうアプリケーション

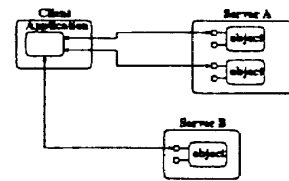


図 2: 分散オブジェクト

を構築できるため、高度なビジネスロジックの処理も可能である。

この際、オブジェクトに定義されているインターフェースを利用してアクセスを行なうため、クライアントはオブジェクト自体の内容は知らなくともインターフェースとその機能、サーバの位置を知っていればオブジェクトを利用することができる。そのため、ネットワーク上に分散されたオブジェクトを目的によって呼び出すことができるので、スケーラビリティも持つ。また、複数のサーバにオブジェクトを登録することができるので、マルチサーバにも対応しており、トランザクションの管理なども行なうことができる。

セキュリティ、マルチプラットフォーム対応については環境によって異なるが、代表的な分散オブジェクトインフラである OMG の提唱する CORBA と Microsoft の DCOM についてみると、CORBA はおよそ 600 社ほどのベンダーが参加しているため、多くのプラットフォームでのサポートが行なわれており、もともと分散オブジェクトインフラとして設計されているため、オブジェクト、トランザクションの管理、セキュリティなどの分散環境の機能は十分に兼ね備えている。DCOM は WindowsNT、Windows95 がメインとなっているが、SoftwareAG 社によって Solaris へのサポートがなされている。DCOM のサービスは、MS Transaction Server を用いることでトランザクションの管理ができるなど、CORBA と同様の分散オブジェクトサービスを有するが、WindowsNT4.0 の段階ではまだ発展途上で、セキュリティなどは整っていない。しかし、次期 WindowsNT5.0 では、ActiveDirectory が提供され、また、セキュリティもケルベロスを用いたセキュリティが実装される予定であり、分散オブジェクト環境としての機能を十分に持つと考えられる。

## 5 結論

インターネット上でスケーラビリティ、耐故障性を持つ高度インテリジェントサービスを行なうためには、データのマルチマスタによるリプリケーションとそのデータベースと連係して高度な分散処理が可能な処理系が必要であることを述べ、その一つの方法として、ディレクトリサービス、分散処理インフラを組み合わせたシステムを構築する方法を述べた。また、分散オブジェクトインフラの例として DCOM と CORBA を挙げ、それぞれが要件をほぼ満たしていることを確認した。

## 参考文献

- [1] N.Shivakumar, J.Jannink and J.Widom, "Per-user profile replication in mobile environments: Algorithms, analysis, and simulation results", Mobile Networks and Applications (2)129-140,1997
- [2] Dr.Rishard Grimes, "Professional DCOM Programming", WROX Press, 1997.
- [3] R.Orfali, D.Harkey and J.Edwards, "The Essential Distributed Objects survival Guide", Wiley, 1996.
- [4] 古山一夫, "DCOM ガイドブック", オーム社, 1997.