

## 複数サーバにおけるユーザセッション分散管理方式

加来田 裕和† 内本 和浩‡ 高木 浩則††  
† 西日本電信電話株式会社 研究開発センタ  
‡ 西日本電信電話株式会社 中国技術総合センタ  
†† 東日本電信電話株式会社 研究開発センタ

### 概要

インターネットは、商取引やオークションなどを行ったりユーザ同士や企業とユーザのコミュニケーションを行ったりする場として発展してきている。筆者らは、それらのサービスを提供する情報流通加工型のシステムのための情報集約・配信フレームワークを構築している。

本稿では、この情報集約・配信フレームワークをベースにして、様々な双方向サービスを行うシステムで必要であるユーザセッション管理方式を提案する。本手法は、ユーザセッションを取得すると同時に、クライアント側で事前に接続するサーバを振り分けておくことと、そのセッションを動的に配置することを特徴としている。また、情報集約・配信フレームワークとユーザセッション管理サービスを用いることによって、更なる生産性向上、システム設計作業の効率化を図ることができる。

## The distributed management method of users' session on the servers

Hirokazu KAKUDA†, Kazuhiro UCHIMOTO‡, Hironori TAKAKI††  
† NTT West Research and Development Center  
‡ NTT West Chugoku Engineering and Technology Support Center  
†† NTT East Research and Development Center

### abstract

The Internet is used as a place of communication of users and companies. Moreover, commercial transaction service and auction service are offered in the Internet. These systems are information gathering, processing, and delivery systems.

The distributed management method for these systems of users' session is proposed in this paper. This method is based on information-gathering / distribution framework. The feature of this method is distributing a user to a specific server and arranging a user session dynamically at the same time it holds the session. Using this method, we reduce the cost and the period of developing information gathering, processing, delivery systems and these constructed systems are easy to design.

## 1. はじめに

現在、インターネット環境が企業だけでなく家庭にまで急速に浸透してきた。これにより、ネットワーク上では様々なサービスが提供されている。例えば、各企業がホームページを開設し、提供する商品やサービスの宣伝を行ったり、多種多様の情報から目的の情報を探し出す検索サービスを行ったり、インターネットショッピングやオークションを行ったりしている。このように、初めは一方向的に情報を発信するだけであったものが、インターネットを利用する人口が増えるにつれ、その情報を基にして取り引きやオークションなど双方向性を生かした、ユーザ同士や企業とユーザのコミュニケーションの場としての利用を行うように発展してきている。

これらのサービスを提供するシステムは、情報を発信し、ユーザなどの応答を集約し、様々な情報を加工し、配信するという一連の手順を追うことによってサービスを提供する情報流通加工型のシステムである。筆者らは、この情報を流通させ加工し双方向のサービスを行うことのできるシステムを安価に構築できる技術が必要だと考え、情報集約・配信フレームワークを提案した[1]。

本稿では、この情報集約・配信フレームワークをベースにして、様々な双方向サービスを行うシステムで必要となってきたユーザセッション管理方式を提案する。

## 2. 情報集約・配信フレームワーク

従来、情報を流し加工するようなシステムを用いて双方向サービスを提供するためには、ソフトウェア開発に対してかなりの投資が必要であった。また、今後ネット上の同様のサービスは増加すると考えられる。そこで筆者らは、容易にカスタマイズを行え安価にシステムを構築できる情報集約・配信フレームワークを提案した[1]。このフレームワークで扱う集約

信対象は、これまでのプッシュ型情報配信技術同様、テキストやHTMLを扱うメッセージ型、ならびにファイルを扱うバルク型であり、以下の3点を考慮したものである。

- 協調動作する複数サーバによるサービス提供。
- 同一サーバ上で複数サービスの同時提供
- 構成管理の容易化

このフレームワークをベースとして利用することにより、情報流通加工型のシステムを効率的に構築できるようになる。

### 2.1. フレームワークの構成

このフレームワークが対象としている情報流通加工型のシステムは、サービスに対するサーバの構成管理をおこなう構成管理サーバが、各サーバへプログラムを送り込むことにより、複数のサーバが協調してサービスを提供するシステムである。サーバでは、図1で示すようなアーキテクチャとなっている。

- A) エージェントレイヤ  
情報集約、加工、配信という一連の動作を行うエージェントが存在するレイヤ。
- B) サービスエレメントレイヤ  
提供するサービスが存在するレイヤ。
- C) サービスエレメント管理レイヤ  
各サービスエレメントを管理するレイヤ。
- D) エージェントプラットフォーム  
エージェントの移動を実現するためのプラットフォームとなるレイヤ。

このアーキテクチャをベースにして、情報集約・配信フレームワークを用いてサービスを構築する。その情報集約・配信フレームワークは、基本エージェントと共有リソースからなっており、以下の機能ブロックで構成されている。(図2)

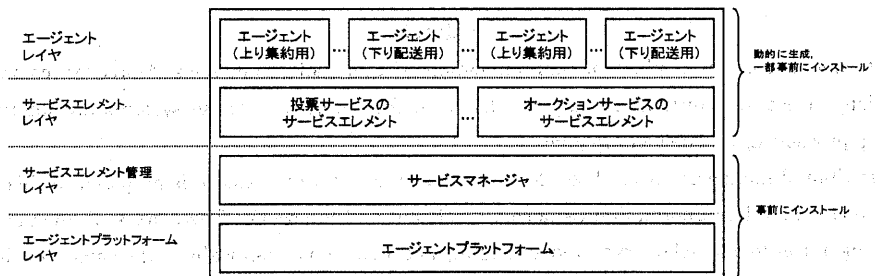


図1. サーバ上のアーキテクチャ

- 制御部
- 集約処理部
- 加工処理部
- 配信処理部
- タイマ部
- 共有情報部

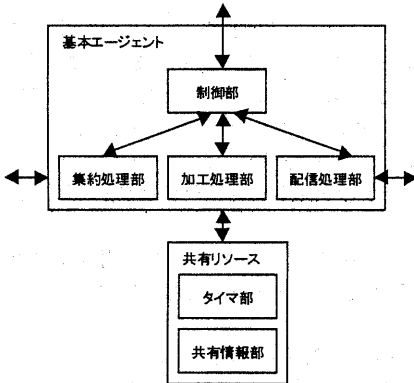


図2. フレームワーク機能ブロック図

## 2.2. 具体的なサービスフロー

あるサービスを提供しようとした場合には、サービスエレメントレイヤにそのサービスを行うエレメントが存在する。たとえば、投票サービスやオークションサービスなどのサービスを行うことができる。さらにそれらのサービスを同時に行うことが可能である。

ここで、具体的なサービス例を基に、サービス提供する場合のフローを考える。ここで念頭において、同期または非同期に行うサービス(表2)を例にとってサービスフローを記述した。

### ● オークションフロー(図3)

オークションでは、各クライアントからユーザ登録、ユーザ認証を経て、オークションサービスへ接続でき、商品に対して指し値を入力する。その後、サービスの終了時点で落札者が決まる。

### ● 投票、アンケートフロー(図4)

投票、アンケートサービスの場合は、クライアントからデータを入力し、サービス終了後データを集計する。

### ● チャットフロー(図5)

チャットでは、クライアントからメッセージを入力して、参加者全員へメッセージを出力する。

表2. サービスの分類

	データを送るのみ	データを送るだけでなく 決済などを伴う
同期	チャット	
非同期	投票、アンケート	オークション

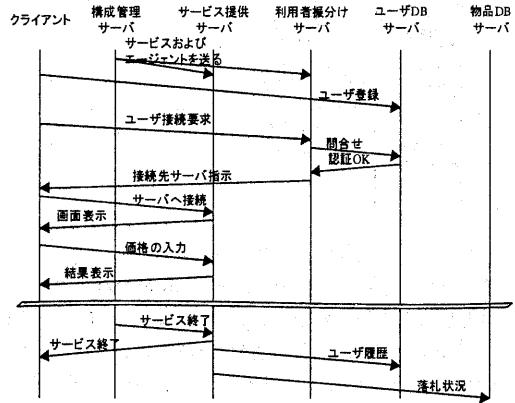


図3. オークションのフロー図

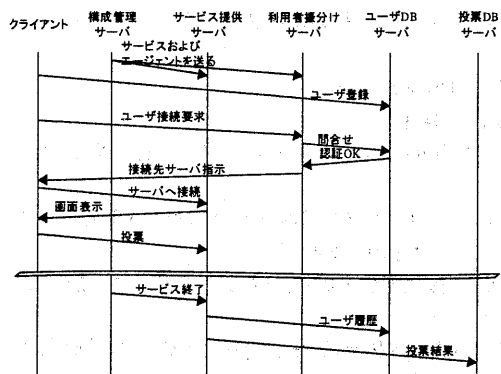


図4. 投票、アンケートフロー図

以上のフロー図を見ると分かるように、それぞれサービスごとのフローには、ユーザ認証などのユーザセッションを取得し管理する一連の作業が共通であることがわかる。基本的には、共有方法が同期か非同期かによってサービスの質は変化すると考えられる。しかし、ユーザセッションの管理という側面から見ると、ユーザの登録、受付、振分け、履歴収集という共通のフローが存在する。このユーザセッションを管理している部分を抜き出し、各サービスへユーザセッション機能を提供できれば、さらに「複数サーバを用いて、容易にカスタマイズを行え安価にシステムを構築」できるようになると考える。また、ユーザが複

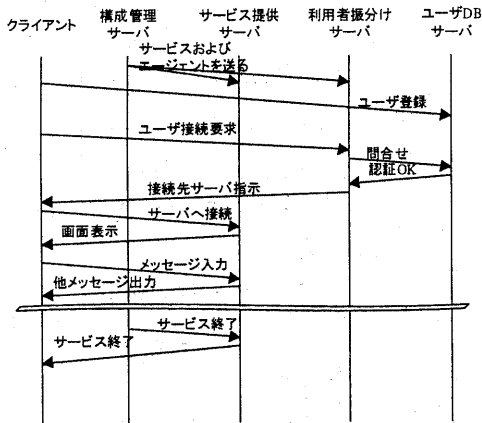


図5. チャットのフロー図

数のサービスを行き来して、その情報を基にサービスを受けることも考えられる。例えば、具体的な議題でチャットなどの意見交換を行っており、その議題について投票やアンケートを行うことがあると考える。非同期サービスと同期サービスのどちらかが一方だけではなく両方のサービスをシームレスに切り替えて利用できるような技術が必要である。

### 3. セッション管理

#### 3.1. 要件

具体的なサービスフローより、下記の事が分かる。

1. 各サービスではユーザセッションを管理することは必須であること。
2. 複数のサービスをまたがって実施することができる。

また、情報集約・配信フレームワークの「複数サーバを用いて、容易にカスタマイズを行え安価にシステムを構築」という目的と考え合わせ、

- フレームワークの用意している情報伝達手段であること。
- 複数サーバで分散してユーザ登録などのユーザ管理ができること。
- チャットや Web コラボレーションのような同期サービスと投票やオークションのような非同期サービスをシームレスに動作させ、ユーザ情報を共有できること。

という要件を満たす、ユーザセッション管理サービスを構築する必要がある。

各サービス共通に使用できるユーザセッション管理サービスを構築することによって、各サービスそれぞれにセッション管理部分を構築するよりも、安

価にシステムを構築することができ、様々なサービスにより使用されることによって、セッション管理サービス自体の品質が向上することができると考える。

#### 3.2. セッション取得方法

クライアントの状態を WWW サーバ側で把握するためには、クライアントを特定しセッションを管理する必要がある。例えば、DB の絞込検索や、オンラインショッピングでのショッピングカートなどのサービスを開発するためには、HTTP のリクエストごとに条件を追加しながら DB 検索したり、サーバ側のテーブルにユーザの状態を保存するなどの方法を取る必要がある。

クライアントを特定しセッションを取得するために、現在利用できるセッション取得方式は次のものである。

1. Cookie を利用する方法。
2. クライアントのブラウザへ HTML を送る際に、FORM 変数へクライアント固有の番号を埋め込む方法。
3. IP アドレスを利用する方法。
4. Java アプレットなどを利用し、特別なセッションを持つ方法。

しかし、Cookie を利用する方法はユーザが受け付けない可能性があり、IP アドレスを利用する方法は、プロキシなどを經由する可能性があるなど要件と照らし合わせると問題である。

要件を考慮した場合には、FORM 変数へクライアント固有の値を埋め込む方法か、Java アプレットなどを利用して特別なセッションを持つ方法が適している。この 2 つの方法により、適用するサービスの範囲から、サービスの中で適宜採用してアクションのトリガとする。

#### 3.3. セッション管理方式

上記のセッション取得方法により、取得したセッションを管理する方式として、基本エージェントをベースにした図6のような機能ブロックを構成した。

1. ユーザデータ受信処理部  
ユーザデータを受信する。また、ユーザセッションを取得する。
2. ID 設定部  
ユーザセッションにより、ユーザ固有の ID を割り振る。
3. ユーザ振り分け処理部  
そのユーザの振り分け先サーバを設定する。

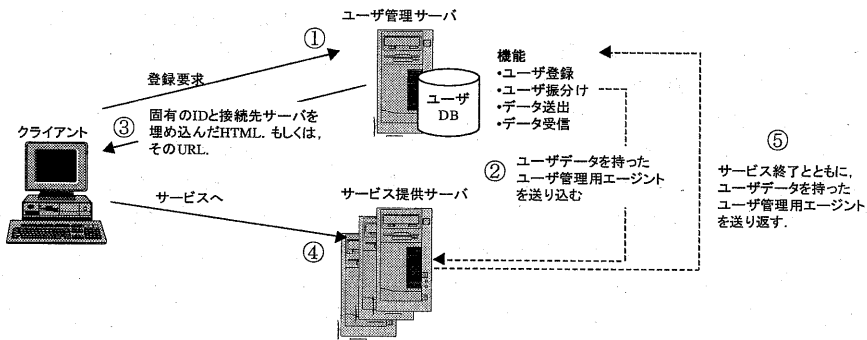


図7. セッションの取得とユーザの振分け

4. ユーザデータ格納処理部  
ユーザデータを DB へ格納する。
5. ユーザデータ送信処理部  
ユーザデータを他のサーバやユーザデータ管理サーバへ送信する。

このユーザセッション管理の特徴は 2 つある。1 つ目の特徴は、ユーザセッションを取得すると同時に、応答データを送信する際にクライアント側で事前に接続するサーバを振り分けておくことである(図7①)。通常、短時間にクライアントからの応答データが集中すると振り分けるためのサーバに負荷が集中し、結果的に処理し切れなくなる応答データが多数発生してしまうという問題が発生するからである。

2 つ目の特徴は、ユーザセッションの動的な配置である。ユーザは、ユーザデータを管理しているサーバでユーザ登録し、認証して各サービスの行われているサーバへ振り分けられる。そのとき、ユーザデータを持ったエージェントは、そのユーザが振り分けられる各サーバへユーザデータを送り込み、ユーザデータを管理する(図7②)。また、ユーザデータを管理しているサーバは、実際そのユーザがサー

ビスを受けるサーバへのリンク情報と、FORM 変数へクライアント固有の値もしくは Java アプレットを埋め込んだ HTML をユーザへ送りこむ。または、その HTML のある URL を提示しておく。ユーザはこの HTML によりサービスをおこなっているサーバへ振り分けられたことになる(図7③)。

各ユーザはサービスが行われているサーバへ接続の際に、ユーザデータを持ったオブジェクトとクライアント固有の値により自動的に関連付けられる(図7④)。セッションの状態ややり取りした情報はすべてこのユーザデータをもつエージェントで管理される。そのサービス(たとえばオークションや投票など)の終了と同時に、またはバッチ処理によりエージェントが管理する更新データをユーザデータベースへ保存し、各サーバのエージェントの管理しているユーザデータは削除する(図7⑤)。

### 3.4. 適用事例

オークションサービスフローを例に、ユーザセッション管理方式を用いて場合を考える。(図8)

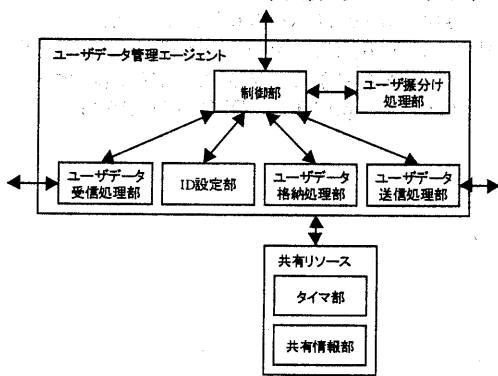


図6. ユーザセッション管理機能ブロック図

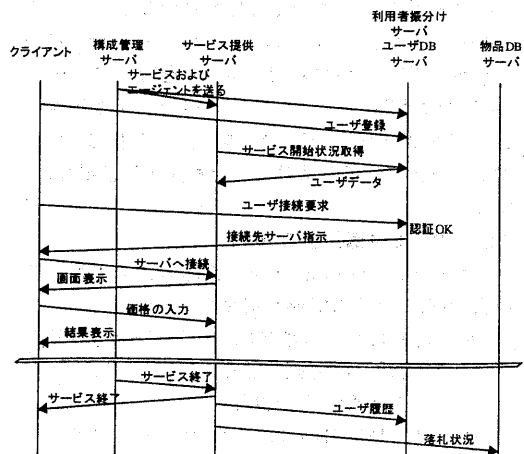


図8. ユーザセッション管理方式によるオークションサービスフロー図

各サービスは、ユーザセッション管理サービスの要求により、各サービスの開始状況を通知する。ユーザセッション管理サービスはユーザ登録の際に、目的のサービスを行っているサーバをユーザへ送り込む。このように、ユーザはユーザセッション管理サービスにより、目的のサービスを提供しているサーバへ接続することができる。また、各サービスを構築する際には、ユーザセッション管理をサービスの一部として構築する必要はなく、ユーザセッションを管理しているエージェントの要求により各サービスの状況を通知すればよい。さらに、各サービスはユーザセッションを管理しているエージェントと通信することにより、サービス相互の関係を保つことができ、シームレスにサービスを提供できるようになる。

## 4. 考察

### 4.1. 有効性の比較

適用事例で述べた以外に、一般的に行われているユーザ管理方式では、セッションを取得した後、アプリケーションや各サービスの内容に合わせて、そのユーザを特定の Web サーバへ固定的に割り当てている。

このように制御することによって、例えばオークションの落札者の落札状態や、商品購入の状況を一元的に管理することができる。しかし、あるユーザデータがあるサーバにしかない場合、そのユーザに提供できるサービスは、そのサーバ上のものだけとなる可能性がある。また、サーバが負荷に耐えられなくなった場合、すぐにサーバを2重化して対応しようとすると、データそのものを移し且つユーザをそのサーバへ接続するように設定しなければならない。

本稿で提案したフレームワーク上のユーザセッション管理方式を使用すれば、ユーザを特定のサーバへ結び付ける必要もなく、サーバを2重化する場合も、図1のアーキテクチャさえ構成してあれば、ユーザデータとユーザの結びつきを考慮する必要もなくなる。

### 4.2. サービス適用上の課題

実サービスへ本セッション管理方式を適用する場合、さらに以下の機能が必要である。

- 停電などの非常時にユーザデータを待避しておき、再起動する際に待避しておいたデータをまとめてくる機能。
- サービス内容に応じて、各サービスのエージェントとユーザデータの管理エージェント

が差分のみをやり取りすることによって、トラフィックを削減する機能。

## 5. おわりに

本稿では、情報流通加工型システムのための情報集約・配信フレームワークを紹介し、このフレームワークをベースにした、分散したサーバ上のユーザセッション管理方式を提案した。このユーザセッション管理方式は、フレームワークの用意している情報伝達手段を持ち、複数サーバで分散してユーザ登録などのユーザ管理ができ、チャットや Web コラボレーションのような同期サービスと投票やオークションのような非同期サービスをシームレスに動作させ、ユーザ情報を共有できる方式である。紹介した情報集約・配信フレームワークとユーザセッション管理方式によるセッション管理サービスを用いることによって、更なる生産性向上、システム設計作業の効率化を図ることができる。

今後は、様々なサービスを提供する際に活用して行き、セッション管理サービスの品質の向上、耐故障性の向上を目指して行く。

## 【参考文献】

- [1] 高木, 日浦, 小泉, フレームワークをベースにした情報集約・配信システム構築方式, 情報処理学会研究会報告, マルチメディア通信と分散処理
- [2] 田淵, 的場, 前野, 阪田, TV 放送に関連付けられたコミュニティ形成支援システム「TV community system」, 情報処理学会研究会報告, グループウェア, pp.49-54, Vol.31, No.9, 1999.
- [3] 山中, Web オブジェクトオーガナイザ, 情報処理学会研究会報告, マルチメディア通信と分散処理, pp.19-24, Vol.93, No.4, 1999.
- [4] 池田, 山田, 長坂, Web 会議間におけるユーザ入力情報の共有方式の検討, 信学技報, オフィスシステム, pp.19-24, QFS98-76, 1999.