

分散型ブラウザにおける 協調動画視聴アプリケーションの開発

青 沼 伴 樹[†] 郭 飛[†] 新 城 靖[†]
佐 藤 聡[†] 中 井 央^{††} 板 野 肯 三[†]

1. はじめに

最近, Web 上での協調作業の手段として, メールやインスタントメッセージといった専用アプリケーションを用いる以外に, Web ベースの協調アプリケーションを用いることが増えている. 一例としては, wiki やブログ, 動画共有サービスなどである. これら Web ベースの協調アプリケーションのフロントエンドとして, Web ブラウザは欠かせないものとなっている.

Web ベースの協調アプリケーションの実行環境として, 分散 Web ブラウザ, あるいは単に分散型ブラウザと呼ばれるブラウザの一種が開発されている¹⁾. 分散型ブラウザは Skype のネットワークを利用し, Remote Procedure Call (RPC) による通信機能を実装している. また, ブラウザ間の Websocket の実装も行われている²⁾.

本研究では, 分散型ブラウザ上で複数人による協調動画視聴を支援するアプリケーションを開発する. また, 分散型ブラウザの通信機能は協調アプリケーションが必要とするグループ通信を行うには適していないため, グループ通信を実現する通信モジュールも合わせて開発する.

これにより, 協調動画視聴のように対話性が高い協調アプリケーションに対して分散型ブラウザが十分な機能と性能を提供することを評価する.

2. 分散型ブラウザの通信機能

分散型ブラウザは, 互いに信頼し合った複数人のユーザによって利用される協調アプリケーションの実行環境である. 分散型ブラウザは, ネットワークで接続された複数のマシン上で, 各マシンにつき 1 つのイ



図 1 通信モジュールの構成

ンスタンスが動作する. それぞれのインスタンスを分散型ブラウザノード, あるいは単にノードという.

分散型ブラウザノードは, インスタントメッセージや Skype の AP2AP API を利用して互いに接続する. 分散型ブラウザ上で動作するアプリケーション同士は, 分散型ブラウザが提供する Remote Procedure Call (RPC) 機能によって通信することができる. さらに分散型ブラウザは, Websocket API によるブラウザ間の通信機能を提供する.

3. グループ通信を実現する通信モジュール

分散型ブラウザが提供する通信機能は, 一対または一対多, つまり 1 つのサーバノードと 1 つ以上のクライアントノードの通信を想定している. しかし, 本研究で開発する協調動画視聴アプリケーションはグループ通信機能を必要とする. 分散型ブラウザの既存の通信機能だけでは, 協調動画視聴アプリケーションの開発が複雑になると予想される. そこで, 分散型ブラウザの既存の通信機能を基盤としてグループ通信機能を実現する通信モジュールを開発する (図 1). 通信モジュールは以下のような JavaScript API を提供する.

```
var group = joinGroup(groupName);
```

joinGroup() 関数は groupName という名前のグループに接続する.

[†] 筑波大学システム情報工学研究科

^{††} 筑波大学図書館情報メディア研究科

```

// Publisher side
var group = joinGroup('player-group');
MediaPlayer.onPlayButtonClick(function() {
    group.publish(
        'media-player',
        {action: 'play'}
    );
});

// Subscriber side
var group = joinGroup('player-group');
group.subscribe('media-player', function(m) {
    if (m.action == 'play') {
        MediaPlayer.play();
    }
});

```

図 2 グループ通信 API の利用例

```
group.publish(topic, content);
```

group.publish() メソッドはグループにメッセージを送信する。topic は送信するメッセージの種類を表す文字列である。また content は送信する任意のオブジェクトである。

```
group.subscribe(topic, callback);
```

group.subscribe() メソッドはグループに送信されたメッセージを受信するためのコールバック関数を登録する。topic は受信したいメッセージの種類を表す文字列である。callback はメッセージを受信したときに呼ばれるコールバック関数である。

これらの API を実装するためには、本研究では、中央のサーバを用いない。その代わりに、分散型ブラウザのノードの1つでメッセージを中継するプログラムを動作させる。

4. 協調動画視聴アプリケーション

協調動画視聴アプリケーションは、協調視聴に参加している複数の分散型ブラウザ上で、サーバから配信される動画を同時に再生する機能を持つ。再生には、一般的な Web ページ埋め込み型動画プレイヤーのインターフェースを用いる。あるユーザが動画プレイヤーの再生ボタンやシークバーを操作すると、その操作が他の分散型ブラウザ上の動画プレイヤーにも適用される。これによって、複数人が同じ動画を同時に、かつ再生操作を同期して視聴することが可能となる。

協調動画視聴アプリケーションから JavaScript インターフェースを介して API を利用する例を図 2 に示す。

MediaPlayer オブジェクトは協調動画視聴アプリケーションの JavaScript インターフェースとす

る。メッセージを送信するクライアントは、まず 'player-group' グループに接続する。そして協調動画視聴アプリケーションにコールバック関数を登録する。このコールバック関数は動画プレイヤーの再生ボタンが押されたときに実行される。コールバック関数は group.publish() メソッドで 'media-player' メッセージを送信する。

メッセージを受信するクライアントは、送信側と同じく 'player-group' グループに接続する。そして group.subscribe() メソッドを実行し、'media-player' メッセージを受信するためのコールバック関数を登録する。このコールバック関数は、受信したメッセージの action プロパティが 'play' のとき、動画の再生を開始する。

5. まとめと今後の課題

本稿では、協調アプリケーションの実行環境としての分散型ブラウザについて述べ、分散型ブラウザ上で動作する協調動画視聴アプリケーションを開発することを述べた。分散型ブラウザが提供する通信機能は協調動画視聴アプリケーションに適していないことを説明し、グループ通信機能を提供するモジュールを合わせて開発することを述べた。

今後の課題は、協調動画視聴アプリケーションを完成させ、分散型ブラウザを評価することである。また、グループ通信モジュールの機能を協調動画視聴以外のアプリケーションでも利用可能にしていきたいと考えている。

参考文献

- 1) Yasushi Shinjo, Fei Guo, Naoya Kaneko, Takejiro Matsuyama, Tatsuya Taniuchi, Akira Sato: A Distributed Web Browser as a Platform for Running Collaborative Applications, The 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2011), (2011).
- 2) 郭 飛, 青沼 伴樹, 新城 靖, 佐藤 聡, 中井 央, 板野 肯三: 分散型ブラウザにおける通信機能の設計, 第 23 回コンピュータシステム・シンポジウム (ComSys 2011) ポスター・デモセッション, (2011).