



ントはサーバと SOAP により通信を行う。サーバは同期のためのデータ (セッション, クライアント識別子, 時計, 状態等) を保持する。クライアントはサーバのデータを操作するメソッドを SOAP を用いて利用する。

クライアントが呼び出すサーバの主なメソッドを表 1 に示す。同期実行のためにクライアントはまずサーバにセッションを作成する (createSession)。セッションは同期実行を行うクライアントのグループを識別するために使用する。またセッションにはクライアントに送るべき命令の待ち行列がある。次にクライアントはそのセッションに参加し (joinSession), セッションにある命令の待ち行列から命令を得る (getCommand)。セッションに参加したクライアントは setPosition メソッドや makeReservation メソッドを用いてゆるい同期に基づく同期実行の命令を送ることができる。クライアントは同期実行のセッションから抜きたいときには leaveSession メソッドを呼び出す。

サーバはあるクライアントから同期実行の命令を受けると、全てのクライアントに命令をマルチキャストしたい。しかし、クライアントサーバモデルではそれが不可能である。この問題を解決するためクライアントは getCommand メソッドを呼び出して命令を受け取るようにする。当初、クライアントはポーリングにより getCommand を呼び出し、命令があった場合のみ命令を受け取っていた (命令がない場合は空の応答を受け取る)。しかし、ポーリングには通信のコストや同期の精度に問題があった。ポーリング間隔を狭めれば精度は向上するが、その分クライアントおよびサーバ両者の CPU の負担や通信トラフィックが増加してしまう。またポーリング間隔を広げればクライアントが命令を受け取るのに遅延が生じてしまい、同期実行の精度が落ちる。そこでポーリングをやめクライアントが getCommand を呼び出すと、命令がない間はサーバは応答を返さず、命令が発行されたときにクライアントに応答が返るといような実装に変更した。

SOAP に対応していないレガシーなクライアントにサーバを対応させるために、HTTP でサーバと通信可能になるようなサーブレットを実現する (図 2)。図 2 では、直接サーバの Web サービスを SOAP で呼び出すクライアントのほかに、HTTP リクエストをサーブレットに発行するクライアントがある。HTTP リクエストを受けたサーブレットが元のクライアントの代わりにサーバの Web サービスを SOAP で呼び出し、元のクライアントには HTTP でレスポンスを返す。

表 1 同期実行のための主なメソッド

メソッド名	説明
createSession(session-name)	セッションを作成
deleteSession(session-name)	セッションを削除
joinSession(session-name)	セッションに参加
leaveSession(id)	セッションから離れる
setPosition(id,milisec,state)	内部時刻を指定する同期実行
makeReservation(id,time,milisec,state)	予約による同期実行
getCommand(id)	命令を受ける

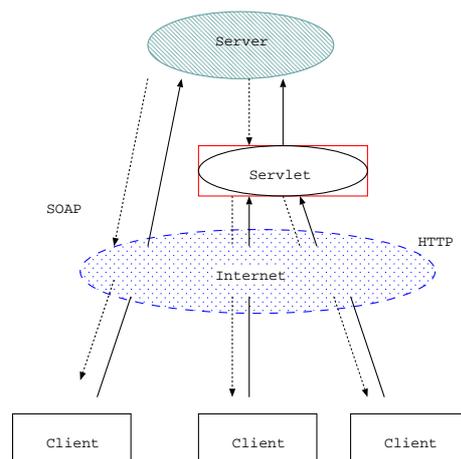


図 2 同期実行システム概要

#### 4. おわりに

本研究ではゆるい同期という概念を用いることで連続メディア演奏プログラムの同期実行を実現する。これまでサーバ全体の仕様を決定し、プログラム設計と開発を行ってきた。現在、サーバの基本的な実装が終わり、テスト用のストップウォッチアプリケーションを作成しサーバの動作テストおよび同期実行のテストを行っている。今後はクライアントを開発するための API のドキュメントの作成をしていく。

本研究の成果は SMART Sync プロジェクトに利用される予定である。このプロジェクトは国立スポーツ科学センター (JISS) のスポーツ情報研究部が、国際競技力向上のために選手とコーチが利用する映像データベースシステムとして開発している SMART System (Sports Movement Archiving and Requesting Technology system)<sup>2)</sup> を発展させるものである。SMART System プロジェクトで開発されている SMART Viewer はサーバの動画像ストリームを XML を用いて検索して再生するアプリケーションである。動画像ストリームに対してはサーバの性能に基づくアクセス制御を行っている。SMART Sync プロジェクトは SMART Viewer に代表されるアプリケーション群を同期実行させるためのプロジェクトである。

#### 参考文献

- 1) MILLS, D.: "Improved Algorithms for Synchronizing Computer Network Clocks.", IEEE/ACM Transactions on Networking, Vol.3, No.3, pp.245-254, June 1995.
- 2) Chikara Miyagi, Koji Ito, and Jun Shimizu : "Creating the SMART system — a Database for Sports Movement ", The Engineering of Sport 6, Vol.3, pp.179-184, 2006.