RESTアーキテクチャスタイルを応用したサーバプッシュプロトコルの提案

吉崎　順太†　　中島　潤‡
北海道情報大学大学院†　北海道情報大学‡

1. はじめに

近年のWebサービスの動向として更新頻度が高いサービスの普及が挙げられる．例えばユーザ間でつぶやきを共有するサービスであるTwitterでは，あるユーザがアクティブな他のユーザ1000人の投稿を閲覧すると１分間あたり約5.4件の新着投稿を受信する．これはブログなど日単位で投稿されるサービスと比べて更新頻度が高い．このTwitterのユーザ数は2009年6月に約4450万人に達し、前年の同月に比べ約15倍増加していると言われており，爆発的に普及が進んでいる．

また，現在のWebにおける標準プロトコルであるHTTP上でサーバプッシュを実現する手法としてCometが登場し，種々のアプリケーションサーバによって実装とAPIの提供が行われている．さらに，それをラップするサーバプッシュフレームワークがオープンソースプロジェクトなどにより開発され提供されている．

Cometを用いたWebサービスの開発において，既存のフレームワークを用いることによりコストが抑えられる．しかしフレームワークを用いた場合，クライアントアプリケーションがそのフレームワークに対して依存性を持ってしまうという問題がある．フレームワークを用いずにサービス独自に実装した場合，どのイベントでどのリソースをプッシュするという対応関係がサービス毎に指針なく定義されてしまうという問題がある．これによりプッシュ機構の再利用が妨げられている．

本研究では，Webサービスにおけるサーバプッシュの振る舞いを一般化し，共通のコンポーネントとして実装可能にすることを目指す．そして，REST アーキテクチャスタイルの制約の導入とHTTPの拡張によりそれを実現し，プロトコルとして提案する．

A Proposal of Server Push Protocol that applies REST Architecture Style
†Junta Yoshizaki, Graduate School of Hokkaido Information University
‡Jun Nakaiima, Hokkaido Information Universitv

2. プッシュ機構の再利用を妨げる問題

Cometにおいてリソースが更新されたときにレスポンスを返すというサービスの振舞いは共通である．しかし，プッシュするリソースとイベントの対応関係がサービスごとに別個に定義されるため，その共通の振舞いのコンポーネント化が妨げられる．例として既存のサービスであるLingr（リンガー）を取り上げる．LingrはCometを用いたWebチャットサービスであり，インフォテリア株式会社が運営していた（2009年5月末をもってサービスを終了しており，現在は利用することができない）．ブラウザ上で動作するWebアプリケーションとしてサービスが開始され，後にHTTP+XMLによるWebサービスが提供されている．そのインターフェイスの一部を以下に示す．

- 発言
  - URL: http://www.lingr.com/api/room/say
  - Method: POST
- 発言の監視
  - URL:http://www.lingr.com/api/room/observe
  - Method: GET

以下，"http://www.lingr.com/api/"をサービスの基底URLとして省略する．Lingrではroom/observeにより新着発言がプッシュされる．プッシュのトリガとなるイベントはあるクライアントからのroom/sayリクエストによる発言である．各クライアントはroom/observeにロングポーリングを行うことで発言の新着通知を受ける．

このroom/observeとroom/sayの２つリクエストの関係はLingr独自のものであり，サービスの実装に依存性をもたらす．このように，Cometによるサーバプッシュにはレスポンスを遅延させるリソースと

更新通知のトリガとなるイベントの関係について何も制約がないため，サービス毎に個別に定義される．

3. サーバプッシュプロトコルの提案

本研究ではHTTPを拡張し，RESTアーキテクチャスタイルの制約を導入することでCometにおけるプッシュの振舞いを一般化することが可能であると考え，プロトコルを提案する．

3.1 RESTによる制約の導入

RESTアーキテクチャスタイルはリソース指向であり，リソースに対して統一されたインターフェイスで操作されるという原則がある．これをWebサービスに適用すると，リソースにURLが与えられHTTPメソッドで操作されるという，実装における制約となる．この制約をCometに導入することで，あるURLに対するPUTおよびPOSTリクエストによる更新をトリガとしてGETリクエストに対しプッシュを行うという一般化した振舞いを定義することができる．

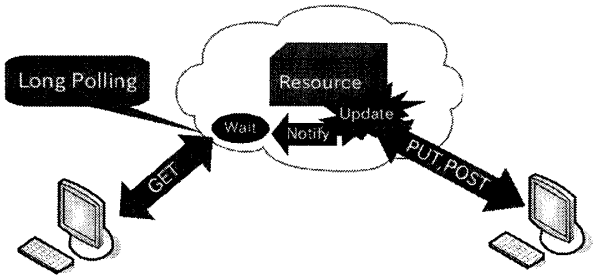

図 1 提案するプロトコルの概要

3.2 条件付きGETの拡張とロングポーリング

HTTPではヘッダにIf-Modified-Sinceフィールドが付与されたリクエストに対し，リソースが指定された時刻以降に更新されていなければサーバは”304 Not Modified”を返すと定義されている．このIf-Modified-Sinceフィールドをプッシュのスイッチとして使用する．クライアントは最初に条件付きでないGETリクエストを送り，サーバは即座にレスポンスを返す．サーバは常にレスポンスヘッダにLast-Modifiedフィールドを付与してリソースの最終更新時刻をクライアントへ伝える．その時刻をIf-Modified-Sinceフィールドにセットし，次のGETリクエストを送る．これに対しサーバはレスポンスを保留し，別のPUTおよびPOSTリクエストによる更新をトリガとしてGETリクエストへ対しプッシュする．以降，クライアントはレスポンスヘッダのLast-Modifiedフィールドの値をIf-Modified-Sinceフィールドにセットして次のGETリクエストを行うという動作を繰り返す．このようにしてロングポーリングが実現する．

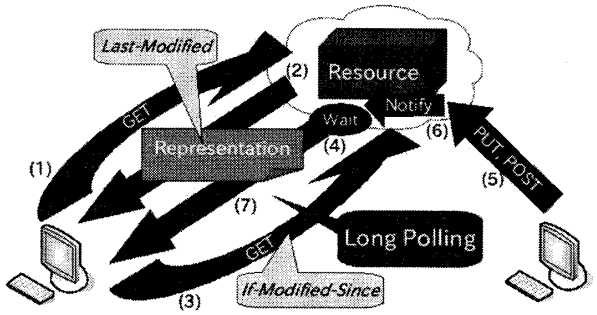

図 2 提案するプロトコルの詳細

4. 提案するプロトコルの利点

先に例に挙げたLingrのWebサービスではプッシュのトリガとなるURLとプッシュされるリソースのURLが別に定義されているため，その２つの関係がサービス固有の依存性を実装に与えてしまうことが問題であった．提案プロトコルにより RESTの制約を導入することで，単一のURLへのHTTPメソッドによるサーバプッシュとして一般化され，サービス固有の依存性を排除することができる．そのため，CometのAPIを提供するアプリケーションサーバにおいて共通のサーバプッシュコンポーネントとして実装し再利用することが可能である．それにより，サーバプッシュを行うサービスを単一のリクエストに対するレスポンス処理に集中して実装可能である．

5. まとめ

本研究ではRESTアーキテクチャスタイルを応用したサーバプッシュプロトコルを提案した．これによりサービス側のプッシュの振舞いが一般化され，共通のコンポーネントとして実装可能になることを示した．

参考文献

[1] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee『Hypertext Transfer Protocol -- HTTP/1.1』(1999), IETF RFC2616.

[2] Leonard Richardson, Sam Ruby (2007.12)『RESTful Webサービス』山本陽平 監訳，株式会社クイープ 訳，オライリー・ジャパン．