

L-027

自律型Webサービス向けの非同期P2Pミドルウェア Asynchronous P2P Communication Middleware

伊東正起† 澤口宗和‡ 大谷真†

Masaki Itou Hirokazu Sawaguchi Makoto Oya

1. はじめに

Web サービスにおいて現状のメッセージングでは対等関係、長期的なセッションの維持においてやや不十分な点が挙げられる。この課題に対し、非同期かつ対等でReliableな通信を行う自律型ミドルウェア (PMAW) を開発した。PMAWはWebサーバの対象配置、セッションの管理の為のLongSessionの定義、データキューにD/Bを用いるなどの方針をとりプロトタイプを開発した。テストを行い、非同期かつ対等型の自律型メッセージングを実装し、本研究の妥当性を確認した。

本論文では2章で研究の背景を述べる。3章で通信プロトコルやセッションの維持についての開発方針を挙げ、PMAW基本構成を示す。4章以降では新たに定義したLongSessionの概念について説明し、方針の具体的な実現方法について述べる。最後にプロトタイプ開発の結果についてまとめ、本研究の妥当性を確認する。

2. 背景と課題

Web サービスを使って企業間での自律的なビジネスメッセージ交換を考えた場合、RPCベースの接続形態では十分ではない。対策としてメッセージングプロトコルについてはWS-Reliability[1]で既に技術確立されている。しかし、通信プロトコルとして用いられるHTTPはクライアント-サーバ型であるため、どちらか一方に主導権を持つ必要があり、ポリシー(タイムアウトや再送方法など)の事前取り決めなども発生する。また、メッセージングにおいても長期にわたるセッションを維持するための不揮発な通信路が必要とされている。このため、非同期かつ対等でReliableな通信を行う自律型ミドルウェア(PMAW)を開発した。

3. PMAWの開発方針

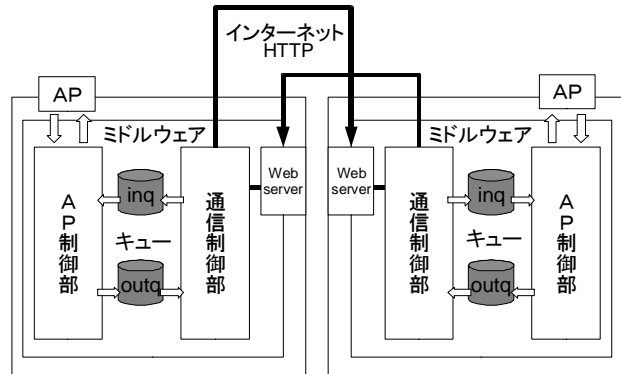
背景で述べた点を考慮し、PMAWでは以下を方針に開発した。

・Webサーバの対象配置

HTTPはクライアント-サーバ型である。対等化するために双方にWebサーバを設置することで下位でのリクエスト-レスポンスを実現する。

・通信プロトコルにSOAPを利用

既に確立されているHTTP等の周辺技術を利用できるように、既存のWebサービス技術に整合させることを方針とした。ここでPMAWの基本構成を図1に示す。



inq : 受信用キュー

outq : 送信用キュー

AP : ユーザの用いるアプリケーション

図1 PMAWの構成

- ・ LongSession について
- ・ 不揮発な通信路の提供

LongSession(以後LS)は長期的なセッション管理を行うために不可欠である不揮発な通信路をミドルウェア上で提供する。ここでの不揮発とはミドルウェアの終了と同時に通信路が終了してしまわないことを指す。LSとセッションを1対1で対応させることにより長期的なセッションを実現する。

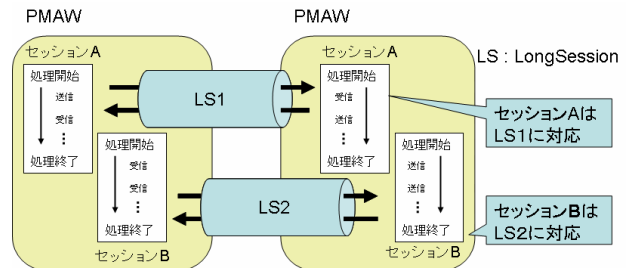


図2 LongSessionとセッション

図2にあるようにLSとセッションを関連付ける。セッションAに対応させたLS1は通信のみを行いセッションAの処理終了まで継続する。LSに関連付けられたセッションも不揮発性を持ち、長期的な運用が可能となる。

・LS通信路の管理

ミドルウェア上での通信路であるためIDを関連付けることにより一意に認識できるようにする。

・LSのIDの決定

LSのIDを決定する際には通信を行う必要がある。その通信は本ミドルウェアの機能を使い非同期に行えるようにする。

・非同期API

PMAWは非同期な通信を行うため、送信・受信の処理にはブロック方式・非ブロック方式の双方を設ける。

† 湘南工科大学

‡ (株)日立情報システムズ

・ 自律的なエラー再送処理

再送処理では処理のエラー状況に合わせて動的に対応を決定する。そのためにエラーの難度レベルを定義しこれを用いて自律的な再送処理を行うことにした。また、送信処理が受信側で失敗した場合、そのときの受信側の都合によって送信側の振舞いを変える事も可能にした。

・ メッセージキューに D/B を利用

PMAW では非同期メッセージング機能を提供するために Store and Forward 方式のキューを介してやり取りを行う。特別なファイル構造を用いず、同期制御や排他処理などのトランザクション処理が重要となるため、キューの実現にはデータベース (D/B) 利用する。

4. LongSession と API

前述したが LS はミドルウェア上で形作られる通信路であり、ID で一意に判別・管理を行う。図3にそのIDの生成の流れを示す。LS の ID は常にユニークである必要があるため、あらかじめ各システムはユニークな情報を準備しておきそれを元に作られる。その情報を特別な ID が割り当てられたあて先が随時変更可能な LS を使い交換する。自分のシステムの情報とあわせ新たな LS に生成規則に従い作られた ID を割り当てる。

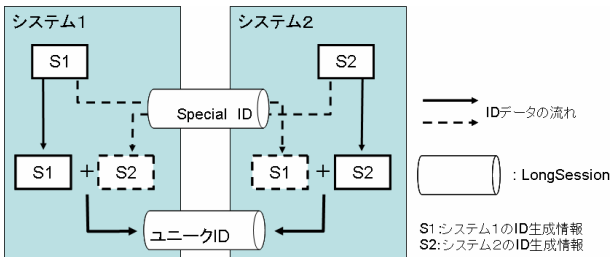


図3 IDの生成

PMAW の API は LS が複数存在していることを考慮してオブジェクトとして提供する。表4にその一例をしめす。

表1 API関数一覧

関数	説明
コンストラクタ	LS の生成
send	非同期送信
check_send	送信結果の確認 (非ブロック)
wait_send	送信結果の確認 (ブロック)
receive	オペレーション指定
check_receive	受信結果および受信データの取得 (非ブロック)
wait_receive	受信結果および受信データの取得 (ブロック)
lsdestroy	LS の終了

5. SOAP ベースの通信プロトコル

SOAP フレームワーク標準[2]により XML のデータを HTTP でやり取りする (表2参照)。送信側から届いたデータに対し受信側は Web サーバから cgi を呼び出しレスポンスを返す (表3参照)。送信側はレスポンスの受信結果を outq に書き込む。一方受信側は inq にデータを格納し、AP 制御部に処理を渡す。

Reliable な通信を満たす為に SOAP プロトコルに独自の情報データを組み込んだ。エラー番号、エラー箇所は自律

的エラー再送処理で使用され、送信失敗通番はダミーデータの番号として通知する。LS 構築の際には受信側の都合も通知する。詳細は7で述べる。

表2 SOAP 送信データ

SOAP ヘッダ部			
lsid	LongSession の ID		
s_ua	自ユーザ名	r_ua	相手ユーザ名
s_url	自 URL	r_url	相手 URL
message_num	メッセージ通番		
SOAP ボディ部			
s_op	自 op 名	r_op	相手 op 名
message_data	ユーザデータ	fault_data	送信失敗通番

表3 SOAP レスポンスデータ

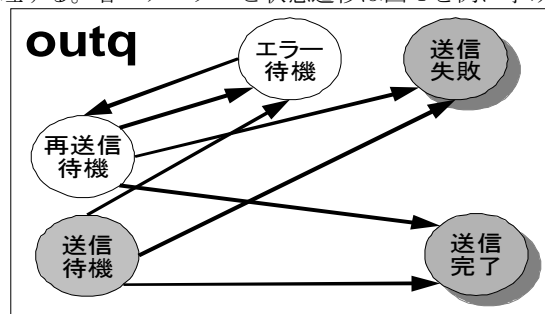
SOAP ヘッダ部			
lsid	LongSession の ID	message_num	メッセージ通番
SOAP ボディ部			
status	送信結果	fault_data	仮登録通番
er_num	エラー番号	place	エラー箇所
retry_param	受信側の都合		

6. D/B によるキューの実現方法

outq は一つのテーブルで構成される。D/B はキュー構造を持たないため、テーブル内に自動インクリメントするシリアル番号のカラムを用意し、データを取り出す際にテーブルソートを実行することで FIFO 構成を実現した。

inq のデータの取り出しは非同期であるため、LS ごとに FIFO なキューが必要である。しかし、LS ごとにテーブルを用意した場合、オーバーヘッドが大きくなる問題が生じる。これを解決する手段として、各レコードデータに LS ごとにシーケンスな番号を付加した。lsid とこの番号でソートを実行することで一つのテーブルでも LS ごとに FIFO なキューを表現することができる。また、このシーケンスな番号は7.2で後述するデータ順序性に用いるメッセージ通番を利用した。

キュー内の各レコードデータは当該レコードの情報として書き込まれるデータ状態を遷移させることによって管理する。各ステータスと状態遷移は図4を例に挙げる。



開始状態：送信待機

終了状態：送信完了、送信失敗

図4 outq の状態遷移図

7. Reliable Messaging の必須要件

7. 1 自律的エラー再送処理

図5に再送機能の流れを示す。sender の送信中にエラーが起きた場合、エラーの発生箇所と種類から待ち時間を決定し再送信をする。HTTP や SQL、ソケットに規定されるコードに対応したレベルを決定する。レベルは0~6までありそれぞれ再送を行う待ち時間の間隔が異なる。再送回数には上限を設け、10回までとした。

再送データは outq にてエラー待機状態となり、エラー箇所とコードが管理テーブルに書き込まれ、レベル判定プログラムが待ち時間を決定し次回送信予定時間が記録される。待機監視プログラムによって予定時間を満たしたデータが outq にて送信待機状態となる。送信が成功するか送信失敗となると管理テーブルのデータは削除される。

受信側 cgi や inq への書き込みの際の sql エラーは相手の都合次第で再送の待ち時間が変化する。よってLSの生成時お互いの再送の都合を送り合う仕組みになっている。

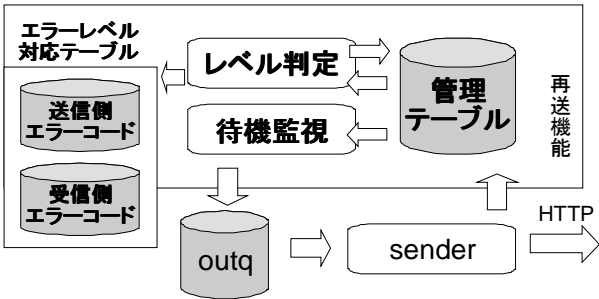


図5 再送機能

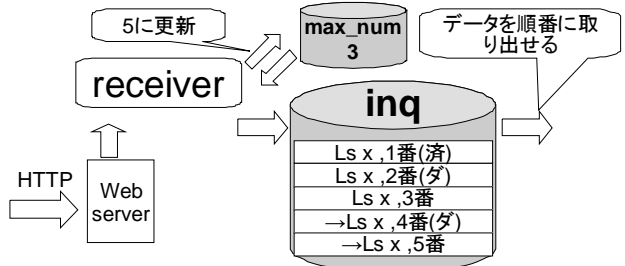
7. 2 データ順序性

通信制御部ではデータの再送信の方針から inq に届くまで、すなわち通信路上ではデータの追い抜きを認めている。しかし6.のキュー構造によりシーケンスな番号さえあればデータの順序性を満たすことができる。そこで AP から送信用データを受け取った AP 制御部が付加する LS にごとにシーケンスな番号(メッセージ番号)を利用する。データを inq に格納した cgi は先ほど格納した LS で inq 内の 0 から数えた連続名メッセージ通番の最大値を max_num テーブルに保管する。これにより最大値より小さな値は順序が保障されたデータ群とみなすことができる。inq から max_num テーブルに保管された最大値以下の値でまだ AP 制御に渡していないデータを取り出し、AP 制御に渡す。

データの中には送信、再送に失敗し相手に送信できないデータが発生する恐れがある。メッセージ通番で順序性を保障する為、このような場合は送信に失敗したデータのメッセージ通番を補完する必要がある。よって失敗リストテーブルという失敗したデータのリストを送信側に用意する。送信側は同 LS の送信データに失敗リストテーブル内の自分と同じ LS のメッセージ通番複数を一緒に送信する。受信側はそのメッセージ通番を用いてダミーのデータを作成し、inq に格納する。送信側はレスポンスから失敗データの格納が出来たかどうかを確認し、成功ならば失敗データのリストから該当データを削除する。これにより見かけ上 inq には通番が穴抜けなく入っているかに見えるのでシーケンスにデータを取り出すことが可能である。

例えば図6のように Long Session X の 1~5 番までのデータを送信する時、2,4 番のデータが送信失敗となった場

合、3番データが2番の、5番が4番の通番を添えて送信され、inq に仮データとして登録される。図6は4,5番のデータを inq に挿入し、max_num テーブルを更新している様子である。



番号:メッセージ通番, (ダ):ダミーデータ
(済):AP に送信済, →:挿入されたデータ

図6 シーケンス保障の処理方法

8. まとめ

本論文で述べた方式を用いて非同期 P2P ミドルウェアのプロトタイプを開発した。テストを行い、非同期かつ対等なメッセージング、エラーレベルを用いたポリシーの動的決定、D/B によるキューの実装、LS による不揮発なセッションの管理など開発方針としていた機能が実装できた。これにより、課題とされた長期のセッションを維持できる非同期かつ対等な自律型メッセージングミドルウェアを実現できた。以上により方式の妥当性を確認した。

9. 参考文献

[1]OASIS, “WS-Reliability 1.1”, OASIS Standard, 2004.
 [2]M. Gudgin et al, SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), W3C Recommendation, 2007.