

AWS における非対称構成型メッセージング機能の実現

吉川 恭平[†] 木村 泰輔[†] 平本 真道[†] 大谷 真[†]

湘南工科大学[†]

1. はじめに

自律型 Web サービス (AWS: Autonomous Web Services) は独立に開発されたシステム間であってもモデル動的協調によりビジネスランザクションが交換できることを狙いとした新たな Web サービス技術である[1]。AWS メッセージ転送は HTTP/XML を用いた非同期メッセージングであり、実現方式検討とミドルウェア実装が行われている[3]。これまでの実装では HTTP 層までは双方のシステムが Web サーバを持つことを前提としていた。しかしこれでは Web サーバを持たない小規模システムが AWS を利用できない。そこで片方のシステムだけが Web サーバを持つような非対称構成にも適用できる新たなメッセージング機能を追加開発した。

2. AWS ミドルウェアとメッセージング機能

AWS ミドルウェアは動的モデル協調層 (MH 層)、アプリケーションフレームワーク層 (AF 層)、自律型メッセージング層 (MS 層) の 3 つの層で構成され [2]、MS 層は非同期のメッセージング機能を AF 層に対し提供している。これまでの MS 層実装 [3] は対称型のシステム構成だけサポートしており、図 1 のようにメッセージを交換し合う 2 つのシステムの両側に Web サーバが存在することを前提としていた。メッセージの送信を行うにはキューにメッセージを格納し、HTTP リクエストでメッセージを送信する。レスポンスが帰ってきたらキューから送信したメッセージを削除する。

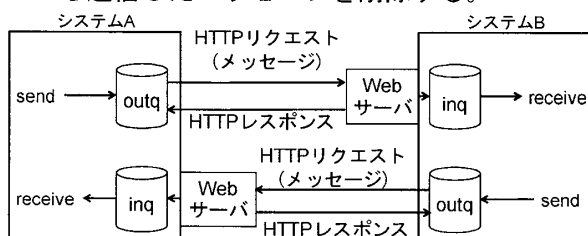


図1 対称型のシステム構成 (従来からサポート)

3. 非対称構成型メッセージング機能と開発方針

SME (Small and Medium Enterprise) や個人、モバイルなどは Web サーバを持たないことが多い。Web サーバを持たないと従来のシステム構成ではメッセージの送信はできるが受信はできない。そこで図2に示す非対称のシステム構成を新たにサポートすることにした。Web サーバが存在するシステムBからWebサーバが存在しないシステムAにメッセージを送信するにはシステムAからHTTPリクエスト (ダウンロード要求) を出し、そのレスポンスでメッセージの送信をすることにした。この方式は ebXML [4] では pull 型と呼ばれ、Implementation of asymmetric type messaging for AWS[†] Kyohei Yoshikawa, Taisuke Kimura, Masami chi Hiramoto, Makoto Oya, Shonan Institute Tecnology

従来の方式は push 型と呼ばれている。なお本論文では Web サーバが存在しない側をシステムA、Web サーバが存在する側をシステムB、HTTP リクエスト (ダウンロード要求) を HTTP リクエストと呼ぶ。

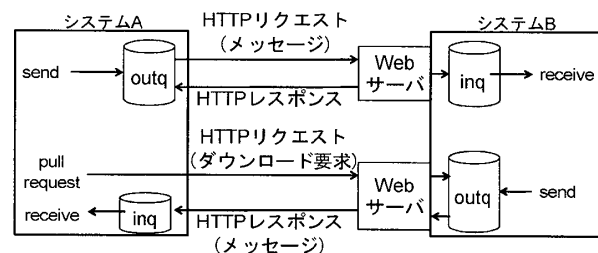


図2 非対称型のシステム構成 (今回新規サポート)

4. 非対称型メッセージング機能の実現

4.1 受信済みメッセージの管理

非対称型ではシステムBからシステムAにメッセージを送信するとき、システムBがHTTPレスポンスが成功したかの判断ができない。このためシステムBはどの時点でキューからメッセージを削除したらよいか分からない。そこでシステムAがHTTPリクエストのメッセージに今まで受信済みのメッセージの連番を格納し、システムBはその連番のメッセージをキューから削除するようにした。

4.2 プロセス/スレッド構造

システムAとシステムBのプロセス構成をそれぞれ図3と図4に示す。システムAはメッセージの送信をするSender、送信が失敗したメッセージの再送管理をするResendHandler、メッセージの受信を管理するReceiverHandler、今回追加したシステムAからシステムBにHTTPリクエストを送信するPullRequesterの4つのプロセスがある。システムBはReceiverHandler、ResendHandler、Sender、の他にメッセージの受信をするためのReceiver、今回追加したHTTPリクエストを受信するためのPullResponderの5つのプロセスに分かれている。またSender、ReceiverHandler、PullRequesterは複数の要求を処理しなければならないことがある。そこで要求ごとに処理が行えるようにマルチスレッド構造とした。

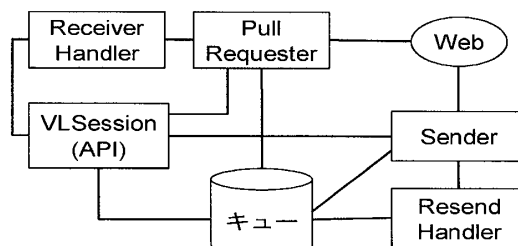


図3 システムAのプロセス構成

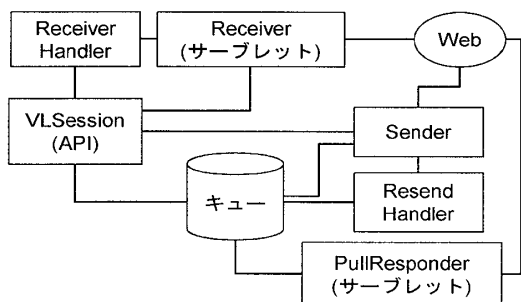


図4システムBのプロセス構成

4.3 API 構造

MS 層の API を表 1 に示す。HTTP リクエストのための pullRequest を追加した。

表 1 API 一覧

メソッド	説明
startVLSession	VLSessionを開始する
restoreVLSession	時停止しているVLSessionを再開する
send	メッセージの送信を行う
receive	メッセージの受信を行う
pullRequest	ダウンロード要求を行う
archiveVLSession	VLSessionを一時停止する
terminateVLSession	VLSessionを終了する

4.4 ミドルウェアの実装構成

非対称型を実装したMS層の構成を図5に示す。PullRequesterはキューから受信済みのメッセージの連番とメッセージ送信に必要な情報を取得しHTTPリクエストを出す。PullResponderはHTTPリクエストがきたら、受信したメッセージからメッセージの連番を取得し削除を行い、キューを調べ、もし未送信のメッセージデータがあれば、HTTPレスポンスでメッセージを送信する。HTTPレスポンスが帰ってきたらPullRequesterはHTTPリクエストを出した際のメッセージの連番をキューから削除し、メッセージデータがあればキューに格納する。

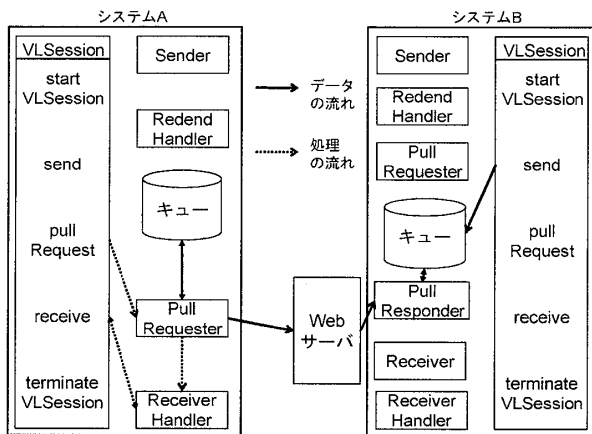


図5ミドルウェアの実装構成

4.5 キュー

従来と同じ信頼性を考慮してキューはDBを使って実装した。図6にそのキュー構造を示す。今回システムAがHTTPレスポンスで受信したメッセージの連番を保持するためのテーブル messagelist を追加した。

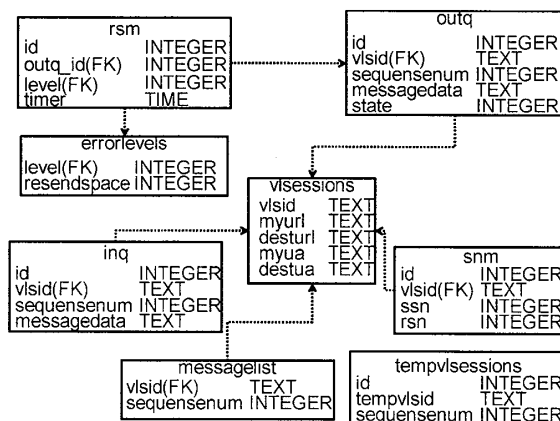


図6キュー構造

4.6 プロトコル

HTTPリクエストとHTTPレスポンスのメッセージの内容を図7に示す。まずシステムAが図7のHTTPリクエストの内容でメッセージを作成し送信する。ボディの sequencenum は今まで受信したメッセージの連番で0個から最大1024個まで格納できる。システムBはHTTPリクエストを受信したら処理をし、図7のHTTPレスポンスの内容でメッセージを作成し送信する。ヘッダの sequencenum はボディの massagedata の連番とセットであり、0個から最大16個まで格納できる。remainingflag はキューにまだボディのメッセージデータ以外に未送信のメッセージデータがあるかを知らせる。

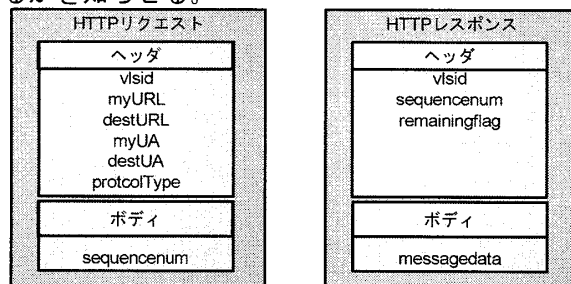


図7プロトコル

6. まとめ

非対称型のメッセージング機能の検討および実装を行った。動作テストを行い非対称型のみの動作テストを行いDBの操作、非対称型のメッセージングの基本動作の確認を行った。本研究は科研費(21500110)の助成を受けたものである。

参考文献
 [1]Oya,M.Autonomous Web Services Based on Dynamic Model Harmonization,IFIP 13E, Toward Sustainable Society on Ubiquitous Networks, Springer,ISBN:978-0-387-8590-2,pp.139 -150,September,2008
 [2]大谷,伊東,塚本,高木,木村,AWS(自律型Webサービス)とそのミドルウェア,情報処理学会第71回全国大会講演論文集,pp.1-503-504,2009
 [3]木村,高木,塚本,大谷,AWSミドルウェアの研究-自律型メッセージング層-,情報処理学会第71回全国大会講演論文集,pp.1-515-516,2009
 [4]OASIS,ebXML Messageing Services Ver. 3.0,Part1, CoreFeatures,OASIS Standard, 2007
 [5]木村,吉川,伊東,大谷,AWSメッセージング基盤改良の検討,情報処理学会第72回全国大会,2010