

## AWS ミドルウェアの研究 -自律型メッセージング層-

木村 泰輔† 高木 良輔† 伊東 正起† 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)の基盤部は、HTTP/XML を用いた非同期メッセージング機能である。昨年度、基本方式検討のためのプロトタイプが作成されたが、標準プロトコル(SOAP、ebXML)の未使用、長期セッションの管理、性能、セキュリティなどが不十分で、これらを解決した本格的な Web サービスメッセージング基盤を開発した。本論文では、AWS メッセージング層の開発方針、設計、実装について述べる。

## 2. AWS メッセージング層とその開発方針

AWS 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つの層から構成される。

MS 層は、AF 層に対し、非同期のメッセージング機能を提供する。MS 層では、VLSession (Very Long Session)という概念を導入し、HTTP 上での非同期メッセージングを実現する。VLSession とは、MS 層内で定義される通信路のことであり、vlsID で一意に判別される。

以下の方針のもと MS 層を開発した。

- (1) 簡潔かつ効率のよいプロセス/スレッド構造
- (2) データベース(DB)を用いた耐久性の高い入出力キューの実現
- (3) store and forward 型の非同期メッセージングの基本機能の用意
  - ・低レベル API
  - ・再送機能など信頼型メッセージング

## 3. プロセス/スレッド構造

MS 層のプロセス構造を図 1 に示す。実線は各プロセスとの関係を表している。通信制御機能をメッセージの送信(Sender)、再送管理(ResendHandler)、メッセージの受信(Receiver)、メッセージの受信管理(ReceiverHandler)の4つのプロセスに分けることにより、各々の制御を効率よく、かつ非同期に処理するような構造にした。Sender はメッセージの送信、ResendHandler は再送となったメッセージの管理、Receiver はメッセージの受信、ReceiverHandler は受信管理を行うプログラムである。各々のプログラムは、DB への

メッセージデータの挿入完了または再送情報を連絡する必要がある。このプロセス間の通信を UDP を用いて実現した。UDP を用いた理由は、再送制御やフロー制御がある TCP より速く通信をすることが可能だからである。この通信は、同一システム内での通信となるため、パケットデータの欠落等がおこる確率が非常に低いと思われる。よって、通信速度の速い UDP を使用した。

Sender と ReceiverHandler は VLSession や Receiver からくる複数の要求を処理しなければならない。そこで、要求ごとに処理を行うようマルチスレッドを用いた。ただし、Sender は送信処理があり一つ一つの要求の処理時間が長くなってしまふ。もし、短時間に多量の要求が発生した場合、リソースを多く消費してしまい処理効率が悪くなってしまふ。この解決策として、スレッドプールを使用しリソースを有効活用できるようにした。

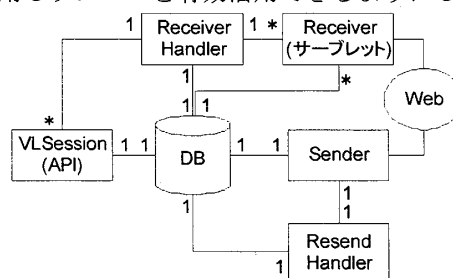


図 1 .MS 層プロセス構成図

## 4. DB の構成

メッセージが長期間キューにとどまる恐れがあり、耐久性の高いキューの実装が必要である。そこで、DB を用いてキューを表現した。キューは送信データを格納する outq と受信データを格納する inq がある。それぞれのキューは、表 1 に示すようにメッセージ、メッセージの通番の他にどの VLSession であるか特定する vlsID、さらに outq には送信状態(未送信、送信失敗、再送)の情報が格納される。

キューの他に、VLSession の情報を保持する vlssessions、vlsID 作成に必要な情報を格納する tempvlssessions、再送する間隔を保持する errorlevels、再送となった outq の id(シリアル ID) や次に再送する時間、エラーレベルを格納する rsm(ReSendManagement)、送信データのメッセージ通番の重複を防ぎ、かつ順番通りに番号を付与および受信データのシーケンスを保障する snm(SequenseNumberManagement)を作成した。

表 1.キューの仕様

カラム	内容
id(PK)	シリアル ID
vlsid(FK)	VLSession の ID
sequensenum	シーケンス番号
messagedata	メッセージデータ
state	送信状態(outq のみ)

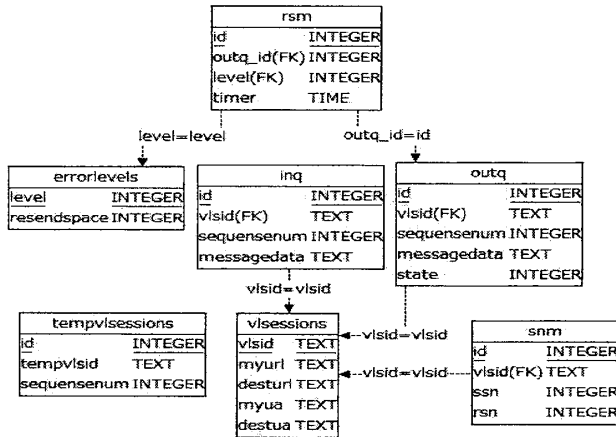


図 2.DB 関係図

## 5. 非同期メッセージングの基本機能と実装

### 5.1 API

MS 層の API は VLSession のクラスメソッドとして表 2 に示す各種機能を実現した。VLSession を開始するには自分と相手の URL と UA(ユーザーエージェント名)が必要となる。UA とはメッセージングのエンドポイントであり、ユニークな名前を持つ。

表 2.API の一覧

メソッド	説明
startVLSession	指定した相手と通信を行うのに必要となる vlsID の作成を行う
restoreVLSession	一時停止している VLSession を再開する
send	メッセージデータを送信する
receive	メッセージデータを受信する
archiveVLSession	VLSession を一時停止する
terminateVLSession	VLSession を終了する

### 5.2 再送機能

メッセージを送信できなかった場合は rsm に格納されている時間まで待機し、Sender に送信要求を出すようにした。ただし、一定回数再送を行っても送信することができなかった場合は、そのメッセージを送信失敗として処理をするようにした。

### 5.3 基本機能の実装

VLSession の一連の流れを図 3 に示す。まず VLSession を特定するための vlsID を作成する。vlsID の作成方法は、お互いが生成した UUID(汎用一意識別子)を交換し文字列比較を行った後、辞書的に大きい方を vlsID とし、セッションを開始させる。メッセージを送信する時は、まず outq にデータを格納し、UDP を使って Sender に連絡をする。Sender は受信したパケットデータから

vlsID とメッセージ通番を取得し、outq からメッセージを取り出す。また、vlsID をキーに自分と相手の UA と URL を vlsessions から取り出しメッセージを送信する。もし送信失敗となったら、ResendHandler に outq の id を連絡する。ResendHandler は次の再送時間まで待機し、VLSession の send メソッドと同じように Sender に UDP を使って連絡をする。Receiver は、メッセージを受信後、inq にメッセージを格納し ReceiverHandler に連絡をする。ReceiverHandler は、VLSession の receive メソッドから UDP で連絡があった場合はその VLSession に対して連絡をし、まだ受信の連絡がなかった場合は何もせず Receiver からの連絡待機状態となる。VLSession を終了するときには、vlsID をキーに全てのテーブルのデータが削除され VLSession が終了する。

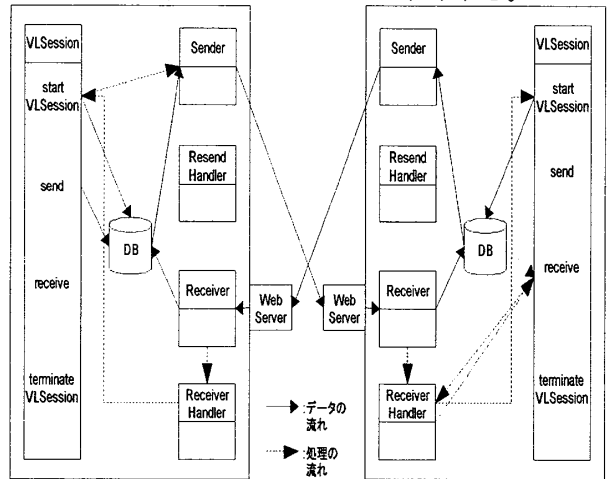


図 3.MS 層の動作

## 6. まとめ

本論文で述べた方式および API・DB の仕様によって AWS メッセージング層の開発を行った。開発環境は、Java1.6.0\_10、PostgreSQL8.3.4、Tomcat6.0.18 を用いた。異なるシステム上で MS 層の動作テストを行い、プロセス/スレッドの動作、プロセス間の連絡、DB の操作、非同期メッセージングの基本機能が正しく動作することを確認した。しかし、VLSession の一時停止/再開、セキュリティ対策がされていない、ebXML の未使用など、不十分な点が残っている。今後、これらの問題点を解決しメッセージング層を完成させる予定である。

本研究は科研費(19500095)の助成を受けたものである。

### 参考文献

1. Simple Object Access Protocol(SOAP) 1.2  
http://www.w3.org/TR/2001/WD-soap12-20010709/
2. 伊東、塚本、高木、大谷、AWS ミドルウェアの研究、情報処理学会第 71 回全国大会、2009
3. OASIS、ebXML Messaging Services Ver. 3.0:Part1、Core Features、OASIS Standard、2007