

AWS メッセージング基盤改良の検討

木村 泰輔[†] 吉川 恭平[†] 伊東 正起[†] 大谷 真[†]

湘南工科大学[†]

1. はじめに

現在、様々な Web サービスの技術の標準化が策定され、ミドルウェア実装製品が登場している。また、Web 上での非同期メッセージングの標準化も完了し、この技術を使用したミドルウェア実装製品が登場しつつある。AWS のメッセージ交換は非同期メッセージングであり、これらの技術をそのまま適用可能である。だが、AWS ではシステム間で運用が統一されていないため、様々な問題が発生することが考えられる。そこで、昨年度から AWS メッセージング基盤の本格的実装を開始した。昨年度は、セッションの維持や耐久性の高い入出力キューの実装を行った。だが、いくつか課題が残っている。

本論文では、これら課題へのアプローチを述べると共に、AWS メッセージング基盤で使われる Web サービスの既存技術の現在の研究状況について触れる。

2. メッセージング基盤の研究状況

2.1. 既存技術

2.1.1. SOAP

SOAP とは、インターネット上で XML データを企業間でやり取りするための通信プロトコルである。最初のバージョンである SOAP1.0 では、HTTP と HTTPS のみでの通信が規定されていたが、最新バージョンである SOAP1.2[1]では SMTP や FTP など様々な通信プロトコルでも利用可能になっている。そのため、ファイアウォールやプロキシサーバを安全に通過することが出来るというメリットがある。

2.1.2. ebXML

ebXML とは、インターネットを利用して企業間取引を実現させるための仕様を定めた標準技術である。通信プロトコルとしては SOAP の普及が進んでいるが、SOAP 単体ではセキュリティやメッセージの信頼性がないという課題が

ある。これら、企業間取引に必要な機能を追加したものが ebXML Messaging Service(以下 ebMS)である。現在の最新の仕様である ebMS version3.0(以下 ebMS3.0)[2]では、SOAP1.2 や WS-Security、WS-ReliableMessaging、WS-Reliability など Web サービス仕様群との互換性を確保している。また、クライアント/サーバ型の通信を実現する Pull 型メッセージングが定められ、ebMS3.0 のコア機能の標準化を完了させている。現在は、このコア機能 (ebMS3.0:Part1, Core Features) を拡張した Part2 の策定中である。

2.2. AWS メッセージング基盤

AWS ではシステム間で運用が統一されていないため、メッセージの送受信に非常に長い時間がかかることが考えられる。そのため、非常に長期にわたるセッション(VLSession と呼んでいる)の維持が必要である。また、VLSession 内でメッセージが長時間キューにとどまることも考えられる。そこで、DBMS などを用いた耐久性の高い入出力キューの実装が必要である。これら問題点を解決するため、昨年度から AWS メッセージング基盤の本格的な実装を開始した [3]。[3]では、以下の方針のもと開発を行った。

- 簡潔かつ効率のよいプロセス/スレッド構造
- 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
- 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

ただし、標準プロトコル(ebXML)の未使用、セキュリティ対策、VLSession の一時停止/再開などが出来ておらず課題となっている。

3. 課題とアプローチ

3.1. ebXML の適用

AWS メッセージング基盤では、ebMS3.0 の通信機能から非同期メッセージングである One-Way/Push メッセージ交換パターン(図 1)と One-Way/Pull メッセージ交換パターン(図 2)の仕様

AWS Examination of messaging base improvement

[†]Taisuke Kimura, Kyohei Yoshikawa, Masaki Ito, Makoto Oya
- Shonan Institute of Technology

に則って開発をしている。その他の機能についても、ebMS3.0 の仕様によって開発を行う予定である。

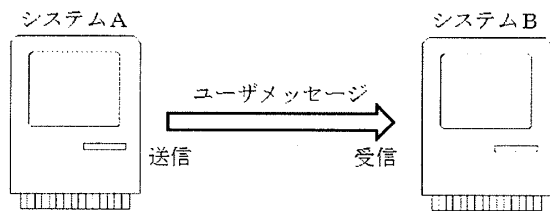


図1. One-Way/Push メッセージ交換パターン

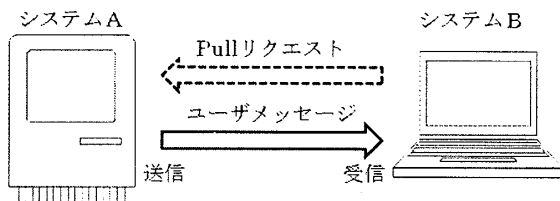


図2. One-Way/Pull メッセージ交換パターン

3.2. Pull 型メッセージング

[3]では、サーバ/サーバ間でのメッセージ交換(これを Push 型メッセージングと呼ぶ)という条件のもと開発を行った。これでは、中小企業や個人などサーバの設置が不可能なところでは、このメッセージング基盤の利用が出来ない。そこで、クライアント/サーバ型のメッセージ交換(Pull 型メッセージング)が出来るようメッセージング基盤の改良を行う必要がある。

Pull 型メッセージングでは、クライアントはサーバに対して自分宛てへのメッセージを受信するために通信相手のサーバに対して Pull リクエストを出す必要がある(図 2)。[3]では、Pull 型メッセージングについて考慮しなかったため、Pull リクエスト等 Pull 型メッセージングに必要な機能や API が用意されていないので、これらを追加する必要がある。

3.3. セキュリティ対策

AWS メッセージング基盤では、以下のセキュリティ対策を講じる必要がある。

- メッセージを送受信する際のメッセージの暗号化
- 通信相手の認証
- DB への接続に関する情報(アカウント、パスワード等)の管理

ebMS3.0 ではセキュリティ機能として WS-Security を採用しており、デジタル署名、メッセージの暗号化などの機能を規定している。本基盤では、この仕様によって準拠させる。

通信相手の認証は、VLSession の開始要求がある通信相手の UA(ユーザエージェント；

メッセージングのエンドポイント)のリストを取得する API を追加し、ユーザが接続待ちの通信相手の認証を可能にさせるようにする。

3.4. VLSession の一時停止/再開

VLSession を一時停止させる場合、バックグラウンドで動いている送受信を行うプロセスに対して、停止させる VLSession の送受信停止命令を出さなければならない。現在の仕様では、送受信を行うプロセスは別々に動いており、これらを管理するものが存在していない。これでは、個々のプロセスに対して停止命令を出さなければならない、またそれらを管理することが難しいと考えられる。そこで、別々に動いているプロセスを管理するマネージャを用意する。VLSession を停止させるときはマネージャに命令を出し、マネージャから各プロセスに対して停止命令を出す。再開させるときも同じように再開命令を出すようにする。これにより、VLSession は一時停止および再開命令をマネージャに対して連絡するだけでよく、マネージャは動作している VLSession の管理が容易になるようになる。

4. まとめ

本論文で述べたとおり、メッセージング基盤にはまだ課題が残っている。これらの課題を解決し、かつ性能の向上を図るにはさらなる研究が必要となるであろう。

AWS ミドルウェアはモデル協調、フレームワーク、メッセージング基盤の各々から必要なものだけを利用できるよう開発を進めている。現在は AWS のメッセージング基盤として動作テストを行っている。今後、メッセージング基盤を使った AWS 以外のアプリケーション等を開発し、このメッセージング基盤の利用性を検証する予定である。

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. Simple Object Access Protocol(SOAP) 1.2 <http://www.w3.org/TR/2001/WD-soap12-20010709/>
2. OASIS, ebXML Messaging Services Ver.3.0: Part 1. Core Features, OASIS Standard, 2007
3. 木村、高木他、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、pp.1-515-516、2009
4. 吉川、木村他、AWS における非対称構成型メッセージング機能の実現、情報処理学会第 72 回全国大会、2010
5. 伊東、塚本他、AWS ミドルウェアの研究－アプローチと構成－、情報処理学会第 71 回全国大会、pp.1-509-510、2009
6. 大谷、伊東他、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、pp.1-503-504、2009