4G－6

Java用 Publish/Subscribe ミドルウエア Secure Distributed InfoBus の設計と実装

浦本 直彦† ・丸山 宏‡
†日本アイ・ビー・エム (株) 東京基礎研究所
‡日本アイ・ビー・エム (株) 東京基礎研究所 および東京工業大学情報理工学研究科

1 はじめに

Secure Distributed InfoBus (SD-InfoBus) は，publish/subscribe (pub/sub) communication のためのミドルウエアである．pub/sub communication では従来の request/response model と異なり，情報の送り手 (producer) は，特定の受け手 (consumer) のアドレスを指定しない．代わりに情報が提供可能であることを示すイベントを生成し publish する．イベントには，情報の特徴を記述した subject が付随しており，その subject を subscribe している consumer がそのイベントを受け取ることができる．これらのイベントの受け渡しは非同期に行われる．この pub/sub に基づくネットワークプログラミングモデルは，Webcasting(Push) の分野で注目を集めており，いくつかの製品も発表されている．

一方，InfoBus [3] は，Lotus と JavaSoft によって提案されている単一 JVM における Java bean 間の非同期，対称型の communication model である．JVM 中の bean は，producer あるいは (かつ) consumer として infobus と呼ばれる object に join することで，data item と呼ばれるデータオブジェクトをやりとりすることができる．producer が，提供するデータが利用可能であることをアナウンスする (fireItemAvaliable という method をコールする) と，consumer 側では，producer が生成したイベントが検知される．この仕組みは JDK1.1 のイベントモデルに基づいているが，InfoBus が提供しているモデルは pub/sub に基づくモデルということができる．

SD-InfoBus は，InfoBus を複数の JVM 内にある bean 間の communication へと拡張したものであり，pub/sub communication を実現するものである．SD-InfoBus の特徴は次の 3 つである．

1. InfoBus との透過性
2. RMTP(Reliable Multicast Transport Protocol [2]) によるマルチキャストのサポート
3. 公開鍵方式によるセキュリティ [1]

本論文では，SD-InfoBus の構成とその仕組みについて述べる．RMTP，セキュリティモデルの詳細は参考文献 [2, 3] を参照してほしい．

Design and Implementation of SD-InfoBus
Uramoto Naohiko† (uramoto@jp.ibm.com) and Maruyama Hiroshi‡ (maruyama@jp.ibm.com)
†IBM Research, Tokyo Research Laboratory. ‡IBM Research, Tokyo Research Laboratory and Dept. of Computer Science, Tokyo Institute of Technology

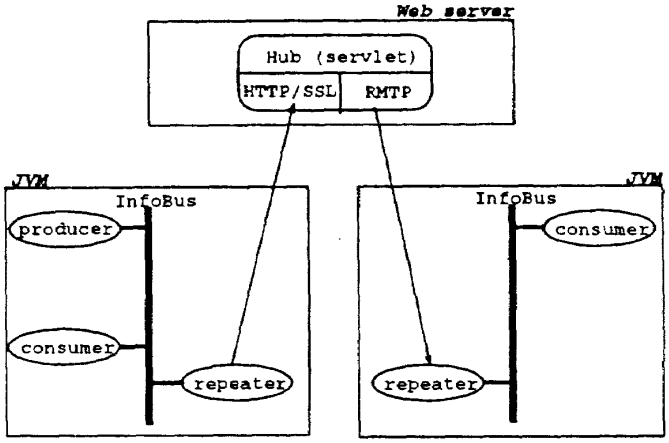

図 1: SD-InfoBus の構成図

2 SD-InfoBus の構成

SD-InfoBus では，repeater と呼ばれるコンポーネントを介して，分散された JVM 内の bean が通信する．repeater は，各々の local infobus に producer/consumer として join し，イベントや data item を中継する．repeater は通信プロトコルとして TCP/IP(peer to peer) と mutlicast をサポートしている．multicast については，低レベルの IP multicast では，データパケットの配送が保証できないため，IBM と NTT が共同開発した RMTP を用いている．その結果，信頼性のある通信を保証しつつ，システムのスケーラビリティを確保することが可能となっている．

RMTP を用いる場合には，repeater は hub と呼ばれる Java servlet に join することで通信が可能になる．hub は，producer からの publish イベントや consumer からの request イベントを join している複数の repeater にマルチキャストする．図 1に SD-InfoBus の構成を示す．

SD-InfoBus では，communication に関する以下のような操作 (API) を提供している．これらは InfoBus の API との高い互換性を持っている．つまり，InfoBus を用いたシステムを容易に SD-InfoBus へと拡張できる．

1. bus および hub への join
2. producer による data item の publish
3. consumer による data item の request
4. data item の更新
5. data item の破棄

```
1: Repeater repeater =
     new Repeater(
      new URL("http://host/servlet/hub"));
2: Consumer consumer = new Consumer();
3: Producer producer = new Producer();
4: repeater.joinInfoBus("ib001");
5: InfoBus bus = repeater.getInfoBus();

6: bus.addDataConsumer(repeater);
7: bus.addDataProducer(repeater);
8: bus.joinInfoBus(producer);
9: bus.addDataProducer(producer);
10:bus.joinInfoBus(consumer);
11:bus.addDataConsumer(consumer);
```

図 2: join するためのプログラム

2.1 Bus および hub への join

図 2に，生成した infobus に，multicast repeater, consumer, producer を join するためのプログラムを示す．SDInfoBus では，peer to peer 接続の場合には 2 つの repeater 同士が，multicast 接続の場合には repeater が hub に join することで，分散した infobus と通信する．local の infobus に接続する producer/consumer は，repeater を意識せず，bus に join することができる (8-11 行目)．hub は，Web server 上の servlet として実現されており，repeater を生成する際に，URL を指定することで，repeater は hub に join する (1 行目)．両者は，Secure Socket Layer (SSL) を用いて相互認証を行う．

2.2 Data item の publish

InfoBus では，DataItem という interface を implement したオブジェクトを bean 間でやりとりすることができる．SD-InfoBus では，さらに Distributable という interface を implement した data item だけが他の JVM 上の bean に配送される．

ある data item を publish したい producer は，その data item に名前 (subject) を付け，join している infobus に対して，それが利用可能であることをアナウンスする．infobus は，イベント (InfoBusItemAvailableEvent) を生成し，各 consumer の dataItemAvailable メソッドを呼び出すことで，イベントを consumer に送る．

イベントを受け取った各 consumer は，data item の名前を確認し，必要であれば，その data item を要求する．SD-InfoBus では，このイベントを次のような手順で他の JVM 上の consumer に配送する．

1. repeater は，consumer として join しているので，InfoBusItemAvailableEvent イベントを受け取ることができる．イベントを受け取ると，producer に data item を要求し，data item を hub に送る．
2. hub は，join している repeater に対し data item をマルチキャストする．
3. data item を受け取った repeater は，local bus に対し，data item が利用可能であることを (producer として) アナウンスする．
4. consumer は，data item が必要であれば，repeater に要求する．

2.3 Data item の request

consumer は，producer が publish した data item を受け取るだけでなく，producer に対して，必要な data item をリクエストすることができる．request イベントは，publish と同じように hub を中継して repeater にマルチキャストされ，repeater がそれぞれの local bus に (consumer として) リクエストする．

2.4 Data item の更新

consumer は，受け取った date item に対し，自分自身を event listener として登録することができる．その data item に何らかの変更があった場合，listener に対してイベントが送られ，データの更新を知ることができる．このような date item 単位での notification は，従来の pub/sub システムにはない機能である．SD-InfoBus では，repeater が local infobus で publish された data item のリスト (item pool と呼ぶ) を管理することで，この機能は実現される．repeater は，local の producer によって publish された data item に対して，自分自身を listener として登録し，item pool に追加する．更新された data item は，hub を通じて他の repeater に送られる．更新された data item を受け取った repeater は，item pool から更新のあった data item を取り出し，そこに登録されている listener に対して，bus を通じてイベントを送る．

2.5 data item の破棄

producer が生成してアナウンスした data item を破棄するには，2.2 節の data item のアナウンスと同様にその旨を infobus に通知する．このイベントは，local bus 上の consumer に送られると共に，repeater および hub を介して別の JVM 上の consumer に送られる．

3 実装と利用例

Hub, RMTP を含めて SD-InfoBus の構成要素のほとんどは Java(JDK1.1) で記述されている．本論文では，JavaSoft から提供されている InfoBus Technical Preview 2 を用いて，SD-InfoBus を実装している．

InfoBus は，JDK1.2 で提供される予定であり，また Lotus は，InfoBus を用いて bean 間で様々なタイプの DataItem (例，表データ) をやりとりすることで，表計算やプレゼンテーションアプリを統合的に提供する eSuite を提案している．SD-InfoBus を使ってこれらを分散環境で制御するといった利用例も考えられる．

参考文献

[1] 丸山宏, 浦本直彦, Java 用 Publish/Subscribe ミドルウエア Secure Distributed InfoBus における鍵配送プロトコル, 第 56 回情報処理学会全国大会, 1997

[2] 高信頼情報分配プロトコル　RMTP, http://www.trl.ibm.co.jp/rmtp/index.htm

[3] InfoBus, http://splash.javasoft.com/infobus/