

## JBossとRed Hat Enterprise Linuxを 活用した証券オンライントレード向け リッチクライアントシステムの優位性

JBoss開発事例  
株式会社 野村総合研究所

株式などのオンライントレード向けシステムでは、たとえば秒間に3万件というような大量のトランザクション処理能力が要求されます。そこで株式会社野村総合研究所様（以下、NRI）では、Red Hat Enterprise LinuxとJBossの製品群を利用することにより、大量の処理を行なえるミッションクリティカルなシステムをローコストで開発されました。そのシステムの構造や特長と、そこで利用されているJBossの諸技術について見ていただければ、同じようなシステムを開発される方にとっても参考になることでしょう。

### リッチクライアント端末で株などを売買

最近では企業の株式などの有価証券を、インターネット経由でオンライン取引する個人投資家が増えてきています。

通常、オンライン取引では、個人投資家がWebブラウザを利用して取引を行うのが一般的です。しかし、この方法では秒単位で刻一刻と変化する株価の最新の時価を知るために、オンライン取引のユーザである個人投資家が都度、各自で情報の再取得（Webブラウザの更新ボタンの押下等）を行う必要があります。これは、操作が面倒な上、わずかとはいえ時間がかかるため、売り時や買い時を逃してしまう可能性があります。

かといってリアルタイムで時価の表示や売買などの操作が行なえる専用のオンライン取引用ソフトウェアを各投資家にインストールしてもらう手法は、配布やバージョンアップの際などの管理が煩雑になってしまいます。

そこでNRIは、リッチクライアントの手法を利用したオンライントレード向けシステムを開発しました。このシステム(図1)では、各投資家は申し込み後の最初のログイン時にのみ、ActiveXのコンポーネントをダウンロードするだけで、Webブラウザを利用するときと同じ間隔で、画面上でリアルタイムな時価の表示や売買が行なえるようになります。

このリッチクライアントシステムでは、それぞれの投資家が注目している銘柄を事前に登録しておく、その時価の情報がリアルタイムで各投資家が見ている画面にプッシュ方式で送信されます。投資家の側から問い合わせや売買の注文が入った場合は、双方向形式での通信が行われます。

株式などのオンライントレードでは、その日の取引が始まった午前9時の直後と、その日の取引が終わる午後3時の直前にアクセスが集中する傾向があります。そこでNRIでは、一人の投資家が見ている画面に30銘柄以上の時価を表示させ、1台のサーバーで1000人の投資家に対応することを想定し、1台のサーバーあたり秒間3万件のトランザクションに対応できるようなシステムを開発しました。

### 秒間3万件に対応するシステムをローコストで

このように株式などのオンライントレードでは、特定の時間帯にアクセスが集中しがちな特性があります。1台のPCサーバーあたり1秒に3万件のトランザクションというのは、かなり高トラフィックなシステムだと言えます。

このような要件を満たすため、高価なサーバー群を大量に導入して大規模なシステムを構築できるのは、事業規模の大きな一部の証券会社だけに限られてしまいます。しかも最近では、そのような証券会社でもシステムの整備や拡張のためのコストを節約する傾向があります。

それに、オンライントレード向けシステムに何らかのトラブルが生じた場合は、大勢の個人投資家などに影響が及んでしまいます。その結果として、その証券会社の信用が低下し、顧客の流出が避けられなくなります。それを避けるためにも、オンライントレード向けシステムには高度な安定性が求められます。

そこでNRIでは、これらの要件を満たすためにRed Hat Enterprise Linuxと、JBoss Application ServerなどのJBoss製品群を活用してオンライントレード向けのリッチクライアントシステムを開発しました。それにより、導入や拡張のために必要となるコストを低く抑える一方、安定性の高いシステムを実現したのです。

その結果として、NRIが開発したシステムは、中小規模の証券会社でも安心して採用できるものとなっています。もちろん大規模な証券会社が、システムの整備や拡張に必要なコストを抑えるために利用することも可能です。そして実際、このシステムを導入して運用している会社の実例も出てきています。



図1：リッチクライアントの画面ショット



### キャッシュにより応答を高速化

フロント側のサーバーには、エンドユーザーからの問い合わせなどに応答するサーバーと、最新の株価の情報をプッシュ方式で配信するサーバーとがあります。

このうち前者のサーバーにはキャッシュを積んでいて、このキャッシュに対してFeedサーバー側から常に最新の情報が配信されてくる仕組みです。そしてエンドユーザーからの問い合わせに対しては普通、Feedサーバーやデータベースとの間でやりとりを行わず、このキャッシュ上のデータを送受信する仕組みになっているのです。この仕組みにより、問い合わせなどに対する処理のうち8割ほどはフロント側で処理できるようになり、やはりFeedサーバー側の負荷が軽減されました。また、ディスクでなくキャッシュ・メモリ上のデータを用いることなどにより、応答の速度も場合によっては数倍に向上します。

このキャッシュの仕組みもNRIでは、独自に開発したものを採用しています。このように「必要な機能は自分たちで自由に開発して組み込むことができる」という点を、NRIではJBoss Application Serverの長所として評価しています。ただしJBossにもJBoss Cacheという仕組みが用意されてきているため、将来的にはそちらへ移行することも検討しています。

また、フロント側のサーバーのうち、最新の株価の情報をプッシュ方式で投資家に配信するサーバーでは、JBossが提供するマイクロカーネルを利用することで、NRIが独自に開発した「Publisher」という仕組みが稼働しています。その仕組みにより、あるポート番号を利用してコネクション・フルの接続を行ないデータを配信するのです。これによって、1秒あたり3万件という大規模なトランザクションが実現されるわけです。このようにミドルウェアの機能をシステムの要件にあわせて拡張できるのは、JBossのマイクロカーネルによるカスタマイズ性の良さの特徴と言えるでしょう。

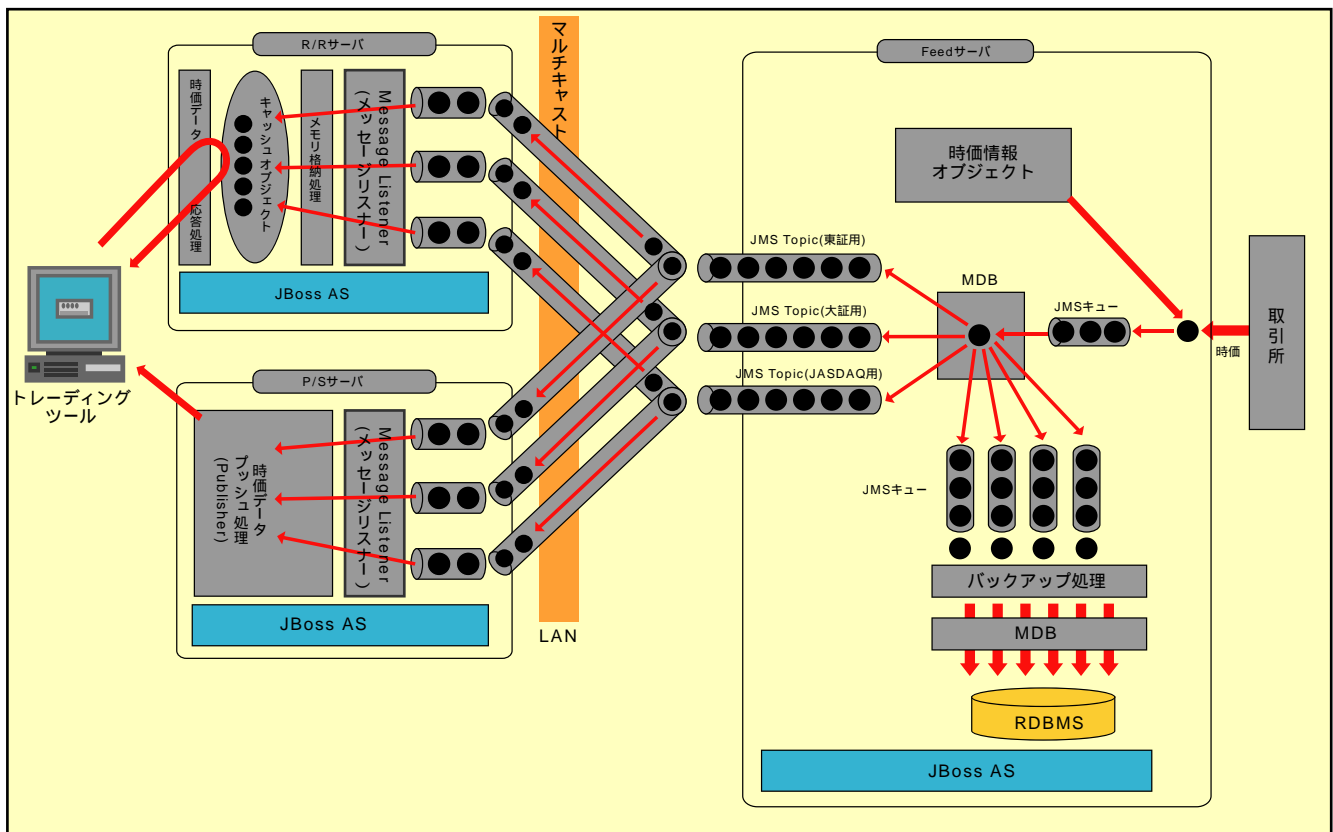
### EJBやクラスタリングで安定性を向上

NRIが開発したシステムでは、JBoss MDB(Message Driven Beans)の機能も利用されています。たとえば、取引所からのリアルタイムデータは万が一の事態に備えて、データベースに書き込む必要が生じます。これを直列で行なうと転送量がパンクする可能性があるため、パラレルでデータベースを起動させパラレルで書き込みを行なうためにMDBを利用しているのです。

Feedサーバーに関しては、JBossクラスタリングの機能も利用されています。Feedサーバーに何らかの障害が発生した結果としてシステム全体の動作が停止してしまうのを防ぐため、Feedサーバーを2重化しています。起動時には2重化された両方のFeedサーバーが同じ動きをし、システムの作動中にはそのうちの片方だけがフロント側のサーバーとやりとりを行なって、もう片方はキャッシュだけを変更して保持する方式をとっています。また、起動時に片方のFeedサーバーだけがフロント側のサーバーにデータを送る形にすると負荷が高くなるので、2重化された両方のFeedサーバーからパラレルでデータが送られる形になっています。

ただし今までのところFeedサーバーのCPUパワーは、通常は2割から3割、日中の取引所からの株価フィード件数が増える午前9時直後でも7割から8割くらいに留まっており、実際に片方のFeedサーバーに障害が発生してももう一方のFeedサーバーに切り替わったという事例は一度も発生しておりません。このようにJBossの各種の機能を利用することでシステムの安全性を高めることができるという点を、NRIでは高く評価しています。「我々の側が最新のパッチを当て忘れたせいでトラブルが発生したことなどはあるが、JBoss製品自体に原因があって問題が発生したことはない」（開発担当者談）。これは、JBossの安定性と信頼性を確認できる点でもあります。

図3：システムのアーキテクチャ



## JBossの構造はサービス指向アーキテクチャ

このようなシステムの安定性や、必要な機能を自由に追加できる点には、JBossソフトウェアの構造が大きく貢献しているとNRIでは見ています。JBossマイクロカーネルに対して必要な機能やサービスだけを自由に選んで追加でき、その起動の順番なども簡単な操作で制御できるからです。「個々のサービスを主体とした構造だという意味で、これはSOA (Service Oriented Architecture) の一つの形だと言えるでしょう」(開発担当者談)。

## SeamなどJBossの今後の機能強化に期待

NRIでは今回開発したシステムのうちマルチキャストを行なわせるための仕組みや、投資家からの問い合わせなどに応じるフロント側のサーバーのキャッシュの仕組みなど、いくつかの部分を独自に開発してきました。しかし、これらの機能に関してはJBoss側でも開発が進められています。

たとえばNRIが開発したフロント側サーバーのキャッシュは、完全なトランザクションを保証しているわけではありません。しかしJBossが開発しているJBoss Cacheでは、キャッシュが完全なトランザクションを実行する機能を持つようになります。この機能に関しては、NRIでも、独自に開発したキャッシュの仕組みからJBoss Cacheへ移行することを検討しています。また、同じように、今後は、JBoss Messagingの利用も検討中です。

また「JBoss Seamを利用すれば安定性の高いシステムを簡単に作れるようになるので、これは最終兵器的なツールになるのではないかと」も期待されています。

## 計画的な開発や情報提供、サポートを計画中

今回のNRIのシステムは、同社の証券オンライントレードシステムの企画開発を担当するインターネットトレードシステム事業部と同社の福岡にある基盤技術四部の方々が共同で開発されました。こちらでは3名の方が、JBossの認定資格をお持ちです。

その方々はコミュニティなどの場で、JBossの技術者と情報交換をしています。「これからもシステムの開発を行なっていく上で、よりいっそうJBossとの連絡を密にしていきたい」とのことです。

またNRIでは、Apache、JBoss、MySQLなどのオープンソースを組み合わせて信頼性の高いシステムインフラを実現するIT基盤ソリューション「OpenStandia」を提供しています。今回のリッチクライアントシステムは、その「OpenStandia」をベースに開発されたものです。

JBossの製品群をレッドハット株式会社が提供していくことになった件に関しては、「一般的にオープンソースのソフトウェアでは、コミュニティによる開発のみであることが多いため、ソフトウェアのリリース時期や品質、万が一のときの対応に不安が残ります。しかし、JBossにおいては、コア開発者の多くがJBossの社員ということもあり、ソフトウェアのリリース時期、品質、問題対応に関してもサブスクリプションという有償サポートを提供しているため安心です。また、レッドハット社に統合されたことにより、これまでレッドハット社が培ってきたオープンソース・ソフトウェアを企業のミッションクリティカルな基幹システムに利用できるようにしてきた仕組みが、これからのJBossにも適用されていくでしょう」(NRI担当者談)。

レッドハット株式会社では今後、JBoss製品群の機能をさらに向上させ、それらを統合して提供していく予定です。また日本国内でのサポートに関しても、より充実させていく計画です。それにより開発者の方々とより一層親密な関係を築いていけるように努めたいと考えています。

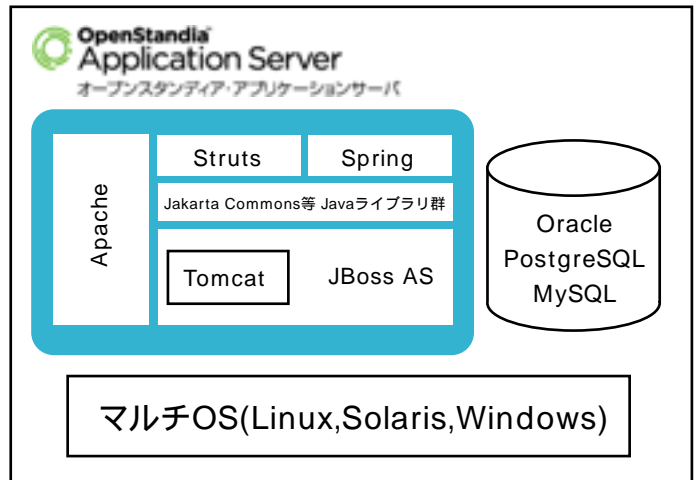


図4 : OpenStandiaの構成図

## < パートナー企業 >

### 株式会社 野村総合研究所

創業 1965年4月1日  
従業員 4,407人 (2007年3月31日現在)  
事業概要 コンサルティングサービス  
ITソリューションサービス  
URL <http://www.nri.co.jp/>

製品やサービスの詳しい内容につきましては、下記URLをご覧ください。



レッドハット株式会社  
<http://www.jp.redhat.com/>

Copyright © 2007 Red Hat, Inc. All rights reserved.  
"Red Hat"、"Red Hat Enterprise Linux"、"JBoss"および"Shadow Man" ロゴは、米国およびその他の国におけるRed Hat, Inc. の登録商標です。Linuxは、Linus Torvalds氏の登録商標です。  
その他全ての登録商標及び商標の所有権は、該当する所有者が保有します。  
本カタログの情報は、2007年3月現在のものです。実際の製品、サービスとは内容が異なる場合があります。