

イントラネットOLTPシステム の実現手法

伊藤 隆行、香川 弘一
(株)東芝 府中工場

4AC-10

1. はじめに

近年の基幹業務（オンライン）システムは、負荷分散のために、各リソースや処理を地理的に分散させることを理想としていた。しかしイントラネットが誕生してからは、開発・運用コストの削減や業務アプリケーションの一括管理などが可能となり、従来の分散システムに替わりイントラネットで基幹業務システムを構築する傾向が強まってきた。しかし現状のイントラネットは、信頼性、安全性をはじめ多数の問題点を抱えている。そこでこれらの問題点解決のため、トランザクションモニタをイントラネット向けに機能強化を行い、イントラネットOLTPシステムを実現した。本稿では、その実現手法を紹介する。

2. イントラネット基幹業務システムの問題点

現状のイントラネットシステムでの問題点をまとめると、以下の4つに大別される。

■C/S型会話処理が困難

現状ではクライアントからの処理の要求都度、セッションの生成／切断が行われる。したがって会話処理を実現させるためには、CGI等を利用し、特殊かつ複雑なプログラミングを行う必要があり、非常に難しい。

■システム全体の信頼性・耐障害性

システムに障害が発生した際のリカバリ処理が考慮されていない。

■性能問題

現状のイントラネットはCGIを使った業務アプリケーションの構築が多い。しかしCGIは処理の要求都度、シェル起動されるため、性能上大きな問題がある。

■システムの拡張性、無停止運用

業務アプリケーションの入れ替え作業が発生した場合、システム全体を停止しなければならない。

3. イントラネットOLTPシステムの実現手法

先に述べた各問題点を解決するためには、トランザクションモニタの導入が望ましい。そこで当社のトランザクションモニタ（TX/ATPS™）をイントラネット向けに機能強化することとした。

TX/ATPS™は、以下の特徴を持っている。

- メッセージリカバリ機能、ジャーナリング機能。
→ システム全体の信頼性・耐障害性の確保
- スケジューリング制御、ロードバランス制御、業務アプリケーションの常駐化。
→ 性能問題の解決
- オンライン構成制御により、システムを停止せずに業務アプリケーションの入れ替えが可能。
→ システムの拡張性、無停止運用の確保

またTX/ATPS™をイントラネットに対応するため、下記のクライアントプラットフォーム別にトランザクション投入のためのAPI（Application Program Interface）を開発した。

●Java版API

TX/ATPS™へ接続／切断、及び業務アプリケーションへのトランザクション（以降：TX）投入用の機能を実装したクラスライブラリを開発した。本APIは、クラスメソッドとして提供され、JavaのソケットによりTX/ATPS™とのセッションを確立する。

●HTML版API

HTML文書からTX投入を実現するAPIを開発した。HTML文書での互換性を確保するため、本開発ではWWWサーバ：Netscape Enterprise Server™^[1]（以降：NSS）の「サーバプラグイン関数」を内部で利用した。しかしHTMLはHTTPプロトコルによりWWWサーバ（NSS）と通信を行うので、処理の度にセッションは切断されてしまう。そこで会話処理実現のために、特別な工夫が必要だった。

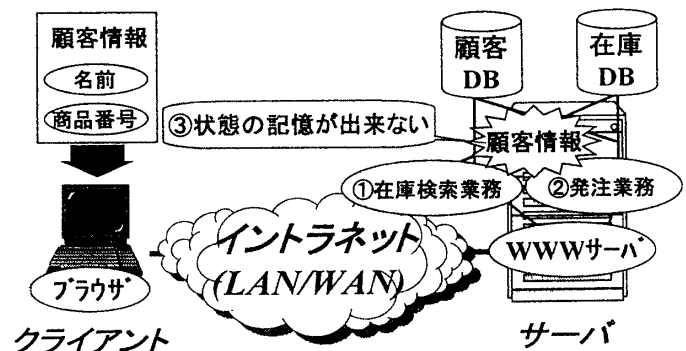


図1：カタログショッピング事例

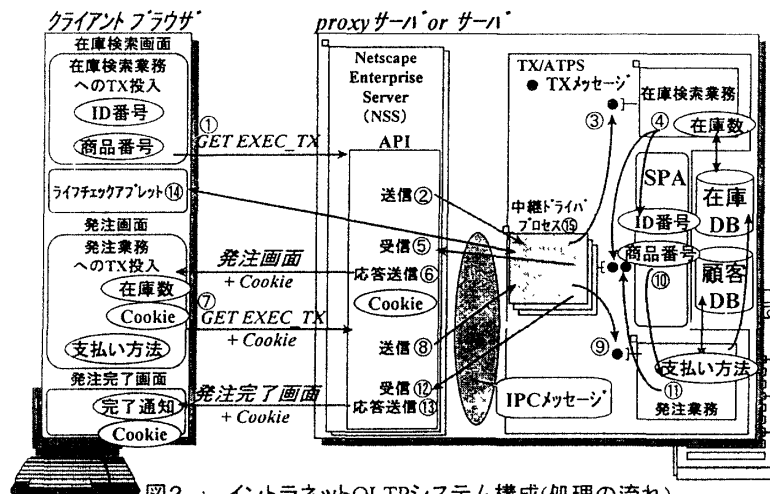


図2: イン트라ネットOLTPシステム構成(処理の流れ)

4. 会話処理の必要性

イントラネット上でのカタログショッピング業務を想定してみると、図1に示すような構成が考えられる。オペレータ（クライアント）は顧客の名前やID番号等の「顧客情報」を入力し、購入要求のあった商品に対し「在庫検索業務」を実行する(①)。応答が返ると、購入するために在庫更新などを行う「発注業務」を実行する(②)。ところが「在庫検索業務」から「発注業務」へ処理が遷移する際、「顧客情報」を記憶しておく必要がある。しかし現状のイントラネットでは、リソースの状態記憶の機構が無い(③)。このように基幹業務を構築する場合、「会話処理の実現」は必要不可欠な設計要素と言える。

5. 会話処理の実現手法

HTML版APIでの会話処理の実現のため、以下の2点の施策を取った。(図2)

(1) セッションの保持機構の開発

クライアントブラウザ（以降：クライアント）とTX/ATPS™間でセッションを保持させるため、下記の手法を取った。

●中継ドライバプロセスの作成

NSSとTX/ATPS™間でセッションを保持するため、常駐ドライバプロセス（以降：中継ドライバ）を作成する。(⑬)

●Cookieの設定処理の作成

APIからクライアントへ応答を返す際、中継ドライバのIPCキー等をCookieへ設定し、TX応答メッセージと共にクライアントへ送信する(⑥)。また設定したCookieは、次のクライアントからの処理要求時に、TX投入メッセージと共に自動的に付加され(⑦)、API内部でIPCキーを参照することにより、再度同一の中継ドライバへの接続が可能となる(⑧)。

●タイムアウトによるセッション解放機能の作成

API実行時から、指定されたタイムアウト時間経過後、強制的にセッションを解放させ、中

継ドライバを終了させる機能を作成した。これはブラウザが開いていてもTX投入処理を行っていない場合、或いはクライアントが存在しない場合に、中継ドライバの残存を防止するために必要である。但しタイムアウト時間経過後も、セッションを保持しておきたい時のために、ライフチェック用のJavaアプレットを作成した(⑭)。これによりタイムアウト時間経過後も、セッションは解放されずに運用が可能となる。

●1ドライバ：nクライアント機能

イントラネットでは、非常に多数のクライアントからのアクセスが多発する。すると中継ドライバも多数生成されるので、サーバの負荷が大きくなってしまいます。そこで1つの中継ドライバに対し複数のクライアントとのセッションを保持させる機構を作成した。

(2) 業務アプリケーション間の情報を共有

TX/ATPS™には、UNIXの共有メモリ機構を利用したSPA（スクラッチパッドエリア）機能がある。本機能により、複数の業務アプリケーションで情報の共有を可能とする。図2の例では、在庫検索業務時に指定された「ID番号」や、「商品番号」をSPAへ格納し(④)、その後の発注業務へ引き渡している(⑩)。

6. おわりに

本報告では、現状構築には問題のあった「イントラネット上での基幹業務システムの実現」というテーマに着目し、トランザクションモニタ（TX/ATPS™）をイントラネット向けに機能強化することで解決した。そしてCGIによる処理よりも高速、かつ高信頼なイントラネット基幹業務システムを実現した。今後は、開発環境、運用環境についても範囲を広げて研究・開発を行いたい。

[1] Netscape Enterprise Serverは米国Netscape Communications Corporationの商標です。