

UNIX 上のトランザクションモニタに基づく帳票処理システムの試作

4 R-8

真鍋 俊彦

関口 幸一

関川 容子

(株) 東芝 総合研究所

1 はじめに

最近、UNIX ベースのダウンサイジングへの期待が高まっている。これはマイクロプロセッサなどの技術的な進歩を背景として、大型計算機で集中処理するよりも、ネットワークで接続された中小型の計算機で分散処理する方がコストパフォーマンスのよいシステムを構築できるようになってきたためである。バッチ処理と並んで重要な処理形態であるトランザクション処理についても、ネットワーク上の複数の UNIX ワークステーションで分散処理できることが求められており、OSF の DCE や UI の UI-ATLAS などの分散処理環境は、その流れに沿ったものである。

しかしながら、分散トランザクション処理システムは未だ開発事例が少なく、システム構築に関するノウハウの蓄積が必要である。そこで、我々はトランザクションモニタ [1] を試作するとともに、業務アプリへの適用実験を進めている。本稿では、この適用実験の一つとして行なった帳票処理システムの構築について、システム構成や得られた知見について報告する。

2 システム構成

2.1 ハードウェア構成

システムのハードウェアは、いずれも当社の計算機である AS4000 シリーズの UNIX-EWS と非 UNIX のミッドレンジマシンである V7070 から構成される(図 1 参照)。V7070 上では従来から集中型の購買用システム稼働しており、人手で作成や承認された帳票は、端末入力によりデータベースに登録される。今回試作した帳票処理システムは、EWS 上で帳票の作成、承認、検索を行い、承認の得られた帳票を V7070 側のデータベースに自動的に登録する機能を有している。

2.2 ソフトウェア構成

システム全体はクライアント・サーバ方式に基づく分散処理システムになっている。トランザクションモニタは、このクライアント・サーバ間に跨るトランザクションの開始や終了のモニタリングなどを行なう。また、このトランザクションモニタは、マルチスレッドサーバに対応し、階層トランザクションをサポートしている。

ソフトウェアの構成を図 2 に示す。EWS 上で動作するプログラムは次の通りである。

クライアントプログラム: GUI(Graphical User Interface) 上でユーザと会話し、帳票データの作成、承認、転送を各種サーバに依頼する

帳票処理サーバ: EWS 側のデータベース (Oracle) に対

し、帳票データの登録、検索、更新などや、ユーザ情報の検索を担当する

通信サーバ: 帳票データのマスターデータベースへの転送時に、EWS と V7070 の間の通信を制御する (ゲートウェイマシン上で動作する)

また、V7070 上のプログラムは次の通りである。

帳票処理プログラム: EWS からの要求に応じて、受信した帳票データのデータベースへの登録などを制御する

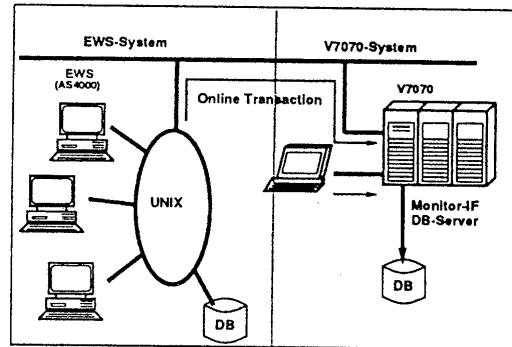


図 1: ハードウェア構成

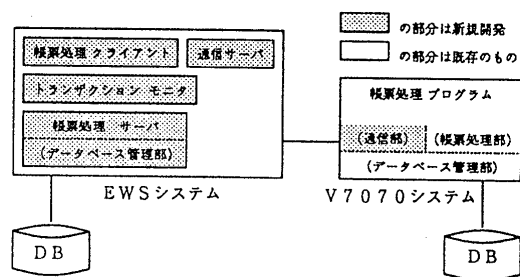


図 2: ソフトウェア構成

3 分散トランザクション処理システムの構築技法

3.1 データベースサーバの機能と構成例

帳票処理サーバは、手配依頼に関する処理を司るサーバで、クライアントから依頼される手配依頼に関する処理(作成、検索など)を行なって、結果をクライアントに

Building an Electorical Form System Based on a Transaction Monitor under UNIX.
Toshihiko Manabe, Koichi Sekiguchi, Youko Sekigawa
Toshiba Corporation R & D Center.

返す。EWS上で作成された帳票はV7070側に転送されるまで、帳票処理サーバで管理される。

このサーバは、トランザクションモニタやクライアントプログラムからの要求をデータベースシステム用の形式に変換するプログラムであり、既存のデータベースシステムの上に付加されたトランザクションモニタ用のインターフェースと考えられる。つまり、帳票処理サーバとはトランザクションモニタやクライアントプログラムからの要求をデータベースシステム用の形式に変換するプログラムである。トランザクションモニタやクライアントプログラムから見れば、帳票処理サーバがデータベースシステムそのものに見える。このサーバプログラムの中で、マルチスレッドや階層トランザクションの構造に応じて、データベースのコミットやアポートを制御している。

3.2 異機種分散処理の実現法

EWSとV7070の間の異機種分散処理はTCP/IP上でシステム固有のトランザクション処理用メッセージを送受信することで実現した。トランザクションのコミットやアポートのタイミングは、EWS上のトランザクションモニタがV7070上のデータベースシステムにこのメッセージを通じて指示を出す。

EWS上では、通信サーバがトランザクションモニタやクライアントプログラムの要求を、この通信インターフェースのメッセージに変換する。EWS上のクライアントプログラムやトランザクションモニタからは、通信サーバがV7070上のシステムに見える。

V7070上の帳票処理プログラムは、受信したメッセージに対応するプログラムを呼び出すようになっている。呼び出すプログラムは、既存の購買システムのプログラムを一部修正したものである。既存のプログラムを流用したので、V7070上の帳票処理プログラムの新規開発分のサイズを小さく抑えることができた。

コミット処理に関しては、使用したデータベースの制限から、まだ、2相コミットをサポートしていない(障害発生時には、データベース間の不整合を防ぐために、更新を取り消すトランザクションを生成している)。片方のデータベースだけでも2相コミットを外部から制御できるようにになったら、その部分だけでも2相コミットに書き換える予定であり、大幅に、不整合を発生する原因を減らすことができる。

3.3 階層トランザクションの利用

帳票処理システムでは、階層トランザクションを活用して、アポート範囲の局所化を計っている。例えば、V7070側でのデータベースの更新に失敗した時には、そのサブトランザクションだけのアポートと再試行を行なう。これにより、回復可能なエラーの場合には、必要最小限の範囲のみ再試行すればすむようになっている。

階層トランザクションが有効であった別の面は、そのモジュール性を活用して、EWS側とV7070側のトランザクションプログラムを柔軟に組み合わせ、システムを構成できた点である。このような記述の容易性はトランザクションが複雑になればなるほど有効である[2]。また、より高負荷なアプリケーションでは、サブトランザクションを並列に実行することで応答速度の向上も期待できる。

4 クライアントプログラムの構成方法

クライアントプログラムは、フォーム制御プログラムと帳票処理用コマンドインタプリタに分割した構成を採用した(図3)。クライアントを構成する二つのプログラムはそれぞれUNIXのプロセスとして協調して動作し、フォーム制御プログラムはユーザの操作をコマンドに変換してコマンドインタプリタへ転送し、その応答をGUI(Graphical User Interface)上に反映する。これらのプログラムは、帳票のフォームやデータ構造の定義記述を読み込んで動作する。帳票フォームの定義記述には、フォームのデザインだけでなく、その振舞いもコマンドインタプリタのコマンドで記述できるようになっている。そのため、フォームの変更や追加は、この記述の変更や追加だけで行なえる。

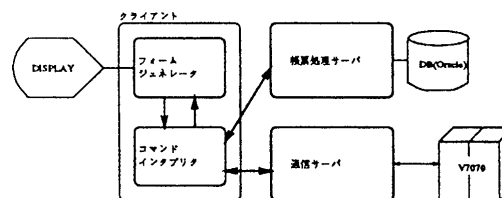


図3: プロセス構成

現在、クライアント側のサーバとのインターフェースは帳票の種類に依存したものになっている。そのため、新しい種類の帳票を追加するには、そのインターフェースを変更する必要がある。この追加作業を容易にするため、このインターフェースをSQLのような標準インターフェースに変更し、上記の帳票の定義記述から帳票の種類に応じたサーバへのメッセージを生成することを検討している。

5 おわりに

UNIX上のトランザクションモニタを利用して試作した帳票処理システムに関して、クライアント・サーバ方式による異機種分散処理の実現方式や、階層トランザクションの活用方式、柔軟性を重視したクライアントプログラムの構成を報告した。

今後はシステムの改良を通して、UNIX上のトランザクション処理システムの構築のノウハウの蓄積を行なうとともに、プログラム開発環境の整備を行なっていく。

参考文献

- [1] 金井、白木原: 階層トランザクション機構によるUNIX上の高信頼分散処理環境、情報処理学会第84回計算機アーキテクチャ研究会(92-8)、1992
- [2] 関口、真鍋、関川: 階層トランザクションを利用した分散アプリケーションの開発、情報処理学会第44回全国大会(1H-2)、1992