

オンラインシステムにおける拡張性を考慮した開始終了処理の高速化

8F-9

松沢 寿典

大石 和寛

菊池 淳一

NTT情報通信網研究所

1. はじめに

バンキングシステムに代表されるオンライントランザクションシステムは、高性能、高信頼性が要求されている。そのプログラム構成は、リアルタイムな汎用的OSと業務プログラム(以下AP:Application Programと略す)とから構成される。こういったシステムでは、APの生産性を向上させるために、データの障害処理等の各種システムに共通的な機能をパッケージングしてAPに提供するDataBase/DataCommunication(以下DB/DCと略す)プログラムを使用することが多い。

このDB/DCプログラムは、資源を管理するモジュール群と処理のシーケンスを制御するモジュール群に大別されるが、後者は他モジュールとのインタフェースを多く持つため、機能拡張時の影響も大きい。

本稿では、シーケンスを制御するモジュールとして、開始終了モジュールを取り上げ、処理の高速性を損なうことなく、拡張性を向上する制御方式について述べる。

2. DB/DCプログラムの構造

(1)プログラムの構造

DB/DCプログラムは、下記の2種類のプログラムに大別される。

[資源管理モジュール群]

各種資源の状態を一元的に管理するモジュール群

[シーケンス制御モジュール群]

開始処理、トランザクション処理といった一連のシーケンスを制御するモジュール群

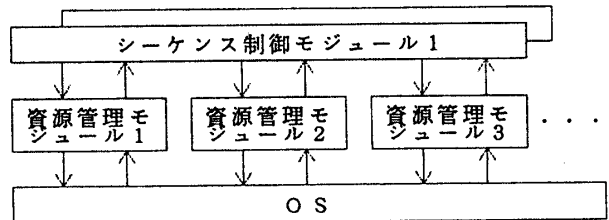
シーケンス制御モジュールと資源管理モジュールとの関係を図1に示す。

シーケンス制御モジュールの例として運転状態を制御する開始終了シーケンスの制御があるが、このモジュールは、資源管理とのインタフェースが多く、性能向上を図るため並列処理が多い。このため、新しい資

High Speed Initializing and Terminating for high Expansion on Online System

Hisanori MATSUZAWA, Kazuhiro OHISHI, Jun-ichi KIKUCHI

NTT Network Information Systems Laboratories



[凡例] → :インタフェース

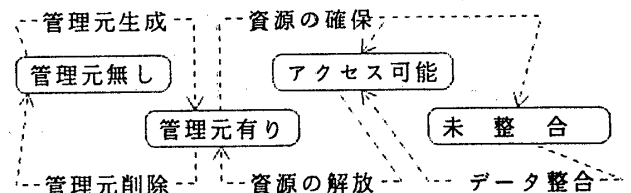
図1 資源管理モジュール群とシーケンス制御モジュール群の関係

源の追加や運転形態の変更といった機能拡張時には、変更が多岐に渡ってしまうという問題点があった。

3. 資源状態の遷移

(1)統一した資源状態の整理

前述の問題点は、資源の状態遷移の統一がとれていないためである。しかし、資源管理モジュール内で個々に管理している細かい状態の遷移に対して、全資源モジュールで統一する事は困難である。そこで、シーケンスを制御するための順序のみに着目し、どの資源管理にも当てはまる大まかな4つの状態に整理した。その状態遷移を図2に示す。



[凡例] □ :状態 --> :処理

図2 資源の状態遷移の整理

基本的にどの資源管理モジュールも開始時には、「管理元が無い」状態から、資源の管理元を生成して「管理元が有る」状態へ遷移し、資源を確保して「アクセス可能」状態へ遷移する。終了時には、開始時と逆の状態遷移になる。また、システムダウン後の開始処理では、障害を復旧するため、「資源元が有る」状態から直接「アクセス可能」状態には遷移せず、「未整合」状態に遷移して、復旧を行った後に「アクセス可能」状態に遷移する。

そして、各資源が管理している状態を上記で整理した状態の遷移に当てはめて、状態遷移の統一を図る事とした。図3に当てはめた例を示す。

例えば、図3の資源1では、S3とS4の状態を併せて、「未整合」状態に対応させる。また、資源3では、「管理元無し」状態と「管理元有り」状態を併せて1つ

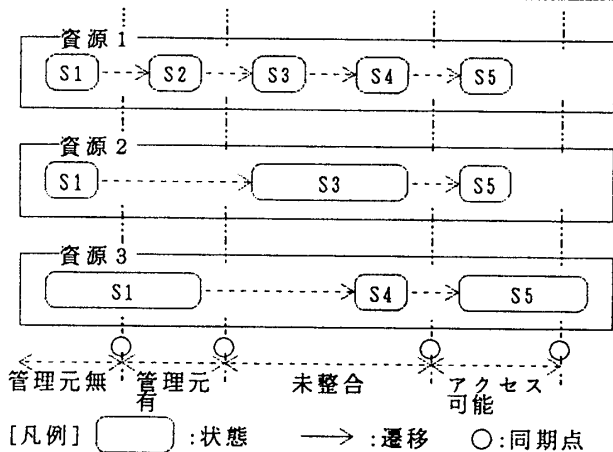


図3 整理した状態遷移への当てはめ

の状態S1として管理していたため、状態を2つの分けた。

(2) 資源の種類の違い

状態の整理は(1)で示したが、資源の種類によって状態を遷移する契機が違って来る。資源の種類としては、DB/DCプログラム自身が使用する資源とAPが使用する資源に分けられ、さらに、APの使用する資源は、2つに分類でき、全部で3つのグループに分類できる。

[DB/DCプログラムが使用する内部資源グループ]

- ・OSが管理している資源であるが、DB/DCとして上記図2のような資源状態を管理しなければならない資源グループで、メモリ、キュー等がある。

[APが使用するデータ整合が必要な資源グループ]

- ・APが使用する資源のうち、障害時にシステムが中断した場合でもデータ内容を整合して、復旧する必要がある資源で、ファイルやデータベース等がある。

[APが使用するデータ整合が不要な資源グループ]

- ・APが使用する資源のうち、データ整合を必要としない資源で、タイマ等がある。

上記の3種類の資源グループについて状態遷移の契機を考えると、例えば「DB/DCプログラム自身の使用する資源がアクセス可能状態にしたあとに、APから使用依頼を受け付ける必要があるため、APが使用する資源グループよりも先に、DB/DCプログラムの使用する資源グループがアクセス可能状態になっていなければならない」のように、それぞれ異なった契機が存在する。

このように、資源の種類毎に状態遷移の契機を決め、これをフェーズと定義した。そして、グループに属する個々の資源モジュールのフェーズ終了時点の同期をとることによって、フェーズ間の独立性を向上させた。表1に開始終了シーケンスにおけるフェーズと状態遷移を示す。例えば、データ整合フェーズでは、APの資

源の整合を行い、アクセス状態に遷移している事を示す。このようにフェーズを定義することで、新たな資源が追加された場合でも資源の種類により状態遷移の契機が即座にわかり、シーケンスの変更も容易である。

(3) 状態遷移の考え方の統一による効果

フェーズを定義し状態遷移の統一を図り、フェーズの終わりには必ず同期制御をすることで、順序性の検討がフェーズ内に限定される。各フェーズを独立した作業工程と位置づけることにより、従来からスケジュール管理等に適用されてきた日程計画法を適用することも容易になり、並列処理が可能であるか明確になる。また、クリティカルパスを明かにすることで、性能向上の際の検討ポイントも明確になる。

表1 開始終了シーケンスにおける資源状態の遷移

使用元	DB/DCプログラム	AP		
		データ整合が必要な資源グループ	データ整合が不要な資源グループ	
開始処理	管理元無	管理元無	管理元無	
	環境1	管理元有	管理元無	
		環境2	管理元有	管理元有
	活性化		未整合	管理元有
		初期化1		
	初期化2		アクセス可能	
		サービス中		管理元有
	終了処理	終了	管理元有	管理元有
		非活性化	管理元有	管理元有
		解放2	管理元有	管理元有
解放1	管理元有	管理元無	管理元無	
	管理元無	管理元無	管理元無	

4. 機能拡張時のシーケンスの変更例

機能拡張として、開始処理をさらに高速化するために、ホットスタンバイ構成で系切り替えを行えるような場合にも、各フェーズが独立していたため、表1のフェーズの活性化フェーズと初期化1フェーズの間に、待機中というフェーズを新たに設定するだけで、他のフェーズのシーケンスに影響なく機能拡張が行える。

5. おわりに

今回述べたように資源の状態遷移の考え方を統一する事で、高速性を損なう事なく拡張性が高い開始終了処理が実現できた。

【参考文献】

[1] T.Nishihara, J.Kikuchi and T.Takehisa : "A DB/DC Platform for Real-time Operating System Based on CTRON Specifications", The Ninth TRON Project Symposium (1992)