B-009

Ｗｅｂシステムにおけるオンラインバッチ処理方式

An asynchronous batch processing method applied in web system

鹿島 啓吾† 引地 一将† 山村 喜恒† 宮崎 肇之†
Keigo Kashima Kazumasa Hikichi Yoshihisa Yamamura Tadashi Miyazaki

１．はじめに

多くの基幹システムでは，利用者と直接対話を行うオンライン処理と大量のデータの一括処理を行うバッチ処理で構成されている。このようなシステムでは，日中はオンライン処理を行い，夜間にバッチ処理を行う。しかし銀行の口座振替業務のように，夜間のバッチ処理とは別に，オンライン処理の実行中に起動するバッチ処理(オンラインバッチ処理)が必要となることも多い。本発表では，Ｗｅｂシステムにおいてオンラインバッチ処理を実現するための考慮点と事例を紹介する。

なお，本稿ではオンラインバッチ処理を「業務的に一括処理であり，オンライン処理の実行中に整合性を保ちながら実行される処理」と定義する。

２．処理方式検討時の考慮点

一般的に，バッチ処理を設計する上で考慮すべき主な点として以下が挙げられる。

・処理性能
・ジョブ運用(スケジュール)
・障害発生時の処理

バッチ処理は大きく分けて２種類あり，１つはセンターバッチ，もう１つはオンラインバッチと呼ばれる。前者は純バッチ，夜間バッチなどとも呼ばれる。オンラインバッチは，オンライン画面中から起動される日中に実行されるバッチ処理という特性上，前述の考慮点に加え，表 1に示す点を考慮する必要がある。

表 1 オンラインバッチ設計上の考慮点

#	考慮点
1	実行処理管理
2	流量制御
3	データの引継ぎ
4	トランザクション管理

以後，表 1に示す点について説明を行う。

1 点目の実行処理管理については，非同期処理として実行した業務処理を管理し，ユーザー側から処理状況が確認できる機能を検討する必要性がある。このとき，監査証跡についても考慮すると良い。これらは顧客から要求される運用要件に大きく影響を受ける。そのためオンラインバッチ処理で実現すべき要件を明確にしておく必要性がある。

センターバッチでは，あらかじめ設定されたスケジュールに従い実行され，データの増加傾向などは予測可能である。突発的に大量データを処理するケースは少ない。しかし，オンラインバッチは利用者からのリクエストに応じて処理を行うため，予期せぬ過負荷となることがありうる。そのため，実行多重度を制御できる方式を検討する必要性がある。

3 点目のデータの引継ぎについては，オンライン側からアップロードしたファイルを入力としてバッチ処理を実行させるように，オンラインで処理しているデータをバッチ処理に引継ぐ必要性があるかを検討する。

最後のトランザクション管理については，オンライン側でロールバックが行われたのにバッチ処理はコミットされてしまうことがないよう，オンライン側とバッチ側の整合性を保つことが求められる。

3. 実装事例について

3.1 要件整理

本節では，2 章を考慮した実装事例を示す。本事例での要件は 表 2の通り。

表 2 要件一覧

#	要件
1	オンライン画面から起動できること
2	実行状況を画面上で随時確認できること
3	利用者の実行履歴が保持されること(監査証跡として使えること)
4	バッチ処理に応じた流量制御ができること
5	起動パラメタ，アップロードしたファイルをバッチ処理の入力として与えることができること
6	オンライン処理と非同期でバッチ処理が実行されること

3.2 処理方式

本節では，3.1 節の要件を満たすための処理方式と検討事項の中で 表 2の 2，3，5および 6 について述べる。本方式では，オンラインプログラムーバッチプログラム間の処理連携，バッチプログラムへの起動パラメタの引渡しはデータベースを介して行う。流量制御は弊社の運用管理製品であるJP1 を利用した。処理概要を 図 1に示し，流れを以下に示す。

① [利用者]オンライン画面上からバッチ起動処理ボタンを押下する。
② [オンラインプログラム]バッチ起動処理を実行する。
③ [バッチ起動処理] 受付番号を発行する。
④ [バッチ起動処理]データベースのステータス管理テーブルに受付番号を主キーとして個人 ID，ステータス，受付日時とともに起動パラメタやアップロードファイルを登録する。
⑤ [バッチ起動処理] 受付番号を含めた起動リクエストをバッチキューに登録する。
⑥ [オンライン処理]バッチ起動処理の結果を画面に返す。
⑦ [バッチプログラム]キューから取り出された起動リクエストの受付番号を元に，バッチ処理の入力情報をデータベースから取得する。(バッチ入力情報取得処理)
⑧ [バッチプログラム]バッチ処理(バッチメイン処理)を実行し，結果に応じたステータスを更新する。

† （株）日立製作所 情報･通信グループ プロジェクトマネジメント統括推進本部, Hitachi, Ltd. Information & Telecommunication Systems. Project & Process Management Division..

⑨ [利用者]ステータス確認画面にて実行状況を随時確認する。

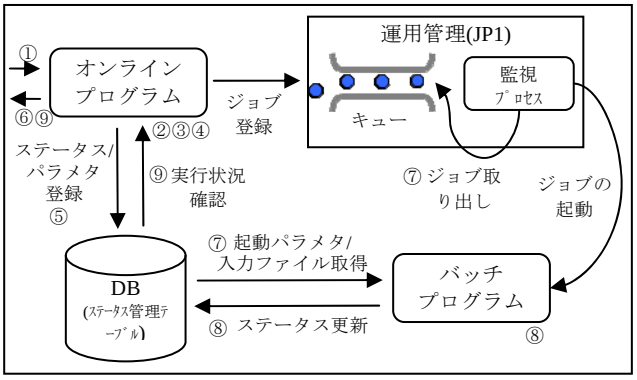

図 1　処理方式概要

本方式では，データベースにてオンラインバッチ処理のステータスを管理する方式とした。利用者は画面上からデータベースのステータス管理テーブルの情報を検索することで自分の起動したバッチ処理の状況を確認することができる。また，起動実績が全て登録されるため，監査証跡としての使用も可能となる。

バッチの入力パラメタとなる実行時パラメタ，アップロードファイルの引継ぎ方式は個別で作り込む部分を削減し，共通化させる方針とした。バッチ処理毎にオンライン側から引き継ぐ起動パラメタ，入力ファイルの有無や数が異なる。図 2に示すように，オンラインからの引渡し処理とバッチでの入力パラメタ解析処理をそれぞれで開発すると保守性も悪く，生産性も低い。

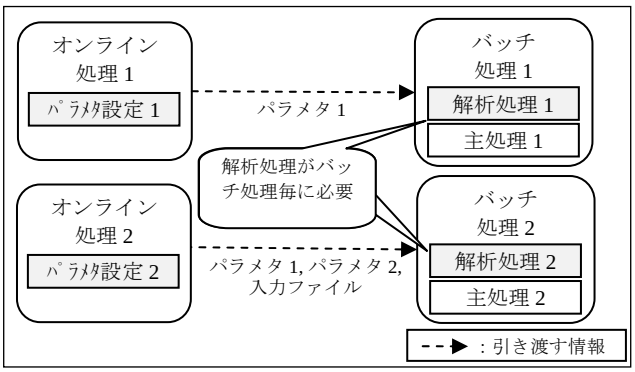

図 2　パラメタ解析処理を各々作る場合

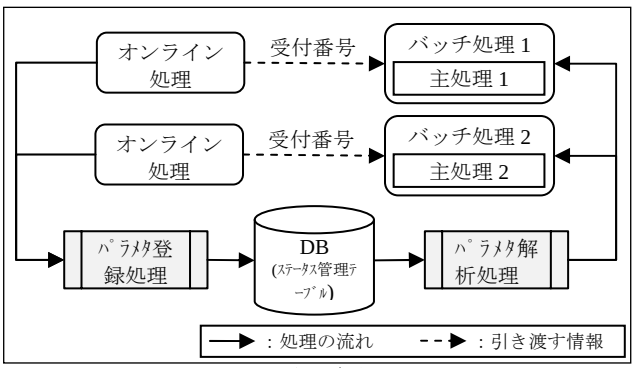

図 3　パラメタ登録・解析処理を共通化

この問題を解決するために，データベースを経由して行う方式に統一させた(図 3)。オンライン側からバッチ側に引き渡す情報を最小限(受付情報のみ)にし，パラメタ登録処理，解析処理にて共通化した。パラメタの数や入力ファイルの有無に関わらず，処理を一元化することで，保守性・生産性を向上させた。

オンライン処理からバッチ処理を非同期で実行させるにあたり，考慮すべき主な問題は 表 3に示す 2 点である。

表 3　トランザクション管理の検討項目

#	検討項目
1	バッチ処理がオンライン側のコミット実行前に実行されるケース。(バッチ処理がオンライン処理を追い越す)
2	オンライン側でロールバックが発生するケース

1 点目のオンライン処理のコミット前のバッチ処理起動については，バッチ処理でステータス管理テーブルへの初回アクセスの際に，トランザクションを同期させる方式とした。ステータス管理テーブルアクセス時に排他ロックをかけることで，バッチ処理はオンライン側のトランザクション確定まで待つ(WAIT)。このためオンライン側のトランザクション確定の前にバッチ処理が起動されても追い越しが発生することはない。図 4にオンライン処理確定前にバッチが起動した場合の処理流れを示す。

2 点目については，オンライン側でロールバックが発生した場合には，バッチ処理はコミットさせない方針とした。バッチ処理側で入力情報をステータス管理テーブルから取得する際，該当する受付番号がない場合は，エラーをリターンすることでバッチメイン処理が実行されないようにした。オンライン側でロールバックが発生した場合，ステータス管理テーブルにレコードがエントリされない状態でバッチ処理が実行される。しかし，バッチ側でステータス管理テーブルを更新しようとした際に該当する受付番号がないため，エラーリターンすることで問題を回避できる。上記のようにトランザクションの整合性を保った。

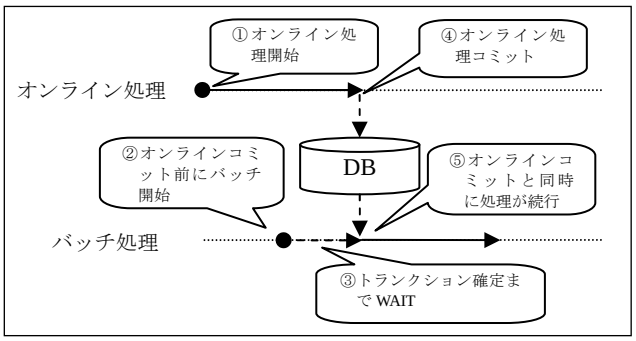

図 4　オンライン処理確定前のバッチ起動

4. おわりに

本稿では，オンラインバッチ処理方式を検討する上での考慮点と実装事例を述べた。2011 年 6 月現在，本稿で紹介した事例のシステムにおいてオンラインバッチ処理の問題は発生していない。今後は本事例の処理方式および共通部品の汎用性を高め，他プロジェクトでも適用するよう検討していく予定である。