

5K-4

履歴カウンタによるオンライン
シンス・インクリメンタルファイルセーブ方式

高桑 勇*, 近藤 純一*, 森 良哉*, 野崎 正治*, 竹歳 明弘**

* (株) 東芝 府中工場, **東芝情報システム (株)

1. はじめに

近年、データ管理は、座席予約システム、発券システムなど対象となる業務内容によって迅速な処理が要求され、データの即時検索/更新、障害発生時のデータ保証などを考慮しオンライントランザクション処理システム (OLTP) の導入が多く図られている。これらのシステムには、平滑な業務の遂行、データの保守などのためにシステム管理者が必要になる。システム管理者には、様々な業務があるが、ここでは、データの保守、特に媒体障害の復旧に関する作業に着目し、オンライン業務下で可能なデータファイルの退避手法について述べる。

オンライン業務は、システム特性上、顧客サービスの面から重要な要因の1つにシステム連続運転 (24H365日) が必要になる。しかし、現状では、データを保守するためにデータファイルを退避媒体へ退避させる場合、システムを一時停止させオフラインで作業しなければならない (オフラインセーブと呼ぶ) 問題点がある。そこで、システム稼働状態で可能な退避手法が望まれる。OLTPで使用するジャーナルリカバリ可能なデータファイルを対象に、退避データの削減が可能なシンス/インクリメンタルセーブ (起点となるセーブからの更新データセーブ機能) を可能としたカウンタ方式によるオンラインファイルセーブ機能 (オンラインセーブと呼ぶ) について述べる。

2. オンラインファイルセーブ

オンラインファイルセーブ機能を実現するに当り以下の項目を課題とした。

- 1) シンス/インクリメンタルセーブをサポートする。
- 2) データファイル退避中には、トランザクションによる更新データを保証する。

ここで、シンスセーブとは、ファイル全体を退避した時点を開始 (トータルセーブ) とし、それ以後更新されたデータだけを退避する方式を意味する。また、インクリメンタルセーブとは、シンスセーブと同様だが起点がトータルセーブだけでなくインクリメンタルセーブも起点とし処理できる方式である。これらのセーブ方式を用いると、トータルセーブだけの運用と比較

し、退避媒体の削減と退避時間の短縮が図れる。このセーブ方式は、一般的にフラグリセット方式により実現される場合が多かった。

2.1 フラグリセット方式

フラグリセット方式とは、データファイル (有編集ファイル) の各ブロックに更新情報フラグを設け、データ更新時に該当ブロックの更新フラグをONにし、データ退避処理を行うとOFFにして更新情報を管理する。このように、更新情報フラグにより容易に更新データブロックだけの退避を行える。しかし、退避処理が完了したブロックの更新情報をリセットするために、データブロックの書き戻し処理が必要になる。そのために、オフラインセーブには有効だが、オンラインセーブには適用できない。

そこで、オンラインセーブを実現するには、書き戻し処理を必要としないタイムスタンプ方式やカウンタ方式が考えられるが、後者を採用した。

2.2 カウンタ方式

カウンタ方式は、データファイルと各データブロックにセーブカウンタを設け、データ更新時に更新対象ブロックのカウンタにデータファイルのカウンタをコピーし、更新ブロックを管理する。図1にカウンタ方式の概要を示す。

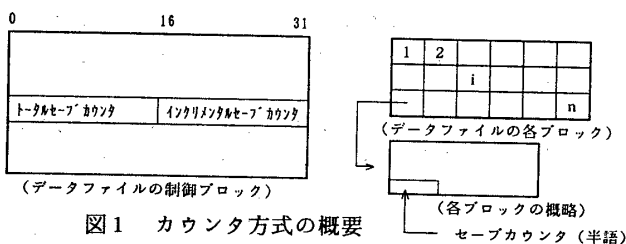


図1 カウンタ方式の概要

本方式では、式 (2.1) に示す条件式により退避するデータブロックを判断する。

$$Sf \leq Sb \quad (2.1)$$

ただし、Sfはデータファイルのセーブカウンタ、Sbは各ブロックに設定されているカウンタ値を意味する。Sfの操作は、セーブツールにより退避処理が行われた時インクリメントする。このように、カウンタ方式で

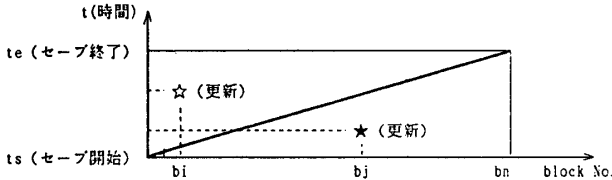
は、データ退避時にデータブロックに関し更新処理がないため、トランザクションによる更新データを破壊せずに処理できオンラインセーブに有効であると考えられ、シンス/インクリメンタルセーブも実現できる。

2. 3 カウンタ方式の性能

カウンタ方式は、フラグリセット方式と比較し性能が向上している。フラグリセット方式は、各退避ブロックに関しread->退避->write 処理を行わなければならないが、カウンタ方式は、read->退避の処理で行えるためディスクアクセスを低減でき、性能的に優れている。

2. 4 カウンタ方式の問題点

しかし、カウンタ方式にも課題2)の問題点がある。これは、図2に示すケースで発生する。



- ★：セーブ処理でブロックを読み込む前に更新されたブロック
(更新後でもデータを退避でき保証できる)
- ☆：セーブ処理でブロックを読み込み後に更新されたブロック
(更新後そのブロックにアクセスせずデータを保証できない)
- ※退避ファイルの全ブロック数をnとする

図2 データファイルセーブ処理の問題

問題とは、ブロック番号biに関して発生する。ただし、biとbjはセーブ処理開始以前に更新されたとする。bjは、退避処理前に実更新が行われているため、退避中の更新処理に対処できる。しかし、biは、退避処理後に実更新が行われたため、退避中の更新情報を保証できない。

このように退避中の更新情報は、退避処理中に退避データとし判断できない場合がある。そこで、これらのデータは、データの更新記録であるアフタイムイメージジャーナルデータ(AIJ)により補うことにした。

以上のことよりオンラインファイルセーブ情報は、本機能により退避した媒体とAIJを合わせて考えることにより保証できる^[1]。

2. 5 大規模データファイルに対する性能向上策

カウンタ方式では、全データブロックを読み込み退避データであるか否かを判断しなければならない。更新データが全ブロックに及ぶ場合には、性能的に問題ないが、更新ブロックが極端に全ブロック数に対し少

ない場合は、不要なread処理が多くなる。この問題には、データファイルの更新情報をビットマップ管理し解決した。ビットマップは、ある単位にデータファイルを分割し、あるビットマップに対するブロックが更新された時に更新情報を設定するようにした。

これにより、大規模ファイルに対しても更新データが少ない場合は、殆どファイルサイズに影響を受けない退避処理を実現できた。

3. オンラインセーブ情報による媒体復旧

実際にオンラインセーブ情報を取得してから、媒体復旧するまでの運用手順を図3に示す。

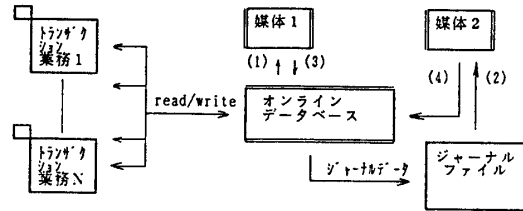


図3 オンラインセーブの運用手順

- 1) データファイル名、または、論理ボリュームを指定しオンラインセーブを行う。
- 2) オンラインセーブが完了したらデータ更新記録(AIJ)をセーブする。

媒体障害が発生したとし、これから媒体復旧処理を行う。

- 3) オンラインセーブによる媒体を用い復旧する。
- 4) AIJの退避データにより復旧する。

4. おわりに

本稿では、システムを停止させることなくデータを退避することが可能なシンス/インクリメンタルセーブを可能としたオンラインファイルセーブ機能の実現方式を述べた。

また、ビットマップを用いることによりファイルサイズにそれほど影響を受けない退避処理を実現している。

(参考文献)

[1] 広兼 他：ミニコンピュータのジャーナルリカバリ高速化手法(5) 第41回情報処理大会論文集