フラッシュメモリを利用する効率的なログ管理と回復処理の実現

4G−6

高倉弘喜　上林彌彦
京都大学工学部

1. はじめに

データベースシステムは障害に備え、定期的にある時刻のデータベースの状態である検査点情報を、また検査点以降のデータ操作を記録したログをそれぞれ二次記憶(ディスク)に保存する必要がある。高速なデータ処理を行なう主記憶データベースでは、主記憶の信頼性が二次記憶より低いため検査点情報とログの保存は重要である。高速なシステムはデータ処理だけでなく回復処理も効率的であることが望ましく、回復時間に大きな影響を及ぼす検査点間隔をできるだけ短くする必要がある。また、従来のログ管理では、データ更新をデータベースに反映する前にログを二次記憶に保存する必要があるため、ログは生成順に保存されていた。

回復処理はデータベース全体の回復後にデータ処理を再開する方式が一般的であるが、データベース容量の増大に比例して回復時間が長くなる問題がある。このため、優先度の高いページから先に回復し、かつ、回復処理をデータ処理と並行して行なう逐次回復が提案された[LEH87]。逐次回復では回復されるページとそのページに対するログだけを読み出すため、ログディスクへの乱アクセスが多発し、回復処理を効率的に行なえない。効率的な逐次回復を行うには、ログをページ毎にまとめてからディスクに書き込む必要がある。しかしログの書き込みが遅れるため、回復処理に利用できないログが存在する問題がある。

本稿ではディスクの代わりにフラッシュメモリを保存媒体に利用することでこの問題を解決し、ログを生成順に保存しても効率的な逐次回復を行なう方式について述べる。また、本稿の方式の性能評価についても述べる。

2. 基本的事項

• フラッシュメモリ

フラッシュメモリは電源供給なしにデータを保持できるEEPROMの一種である。本稿のシステムではNAND型フラッシュメモリを利用する。アクセスは図1に示すように内部レジスタ(256B)を介して行なわれる。また、フラッシュメモリにデータを書き込む前に、チップ全体あるいはページ単位の消去を行なう必要がある。

レジスタのアクセスサイクルは数十から百ns程度である。レジスタとメモリセルのデータ転送は読み出しに15μs、書き込みに35μsかかる。また、消去はページ消去で6ms、チップ消去で10msかかる。データ転送や消去時間はディスクのアクセス待ち時間に相当すると考えられるが、順アクセスでも乱アクセスでも同じである点がディスクとは大きく異なる。

• 従来のログ管理方式

従来のログ管理はログを生成順にディスクへ保存するため、逐次回復を行なうと多くの乱アクセスを生じる。逐次回復の効率化のため以下の示すログ管理法が提案された[LEH87]。ログは生成されると、ページ別に分けられ一時的に不揮発メモリに保存される。各ページごとのログはある程度集まるとディスクに書き込まれる。更新は不揮発メモリにログを書き込んだ後にデータベースに反映される。

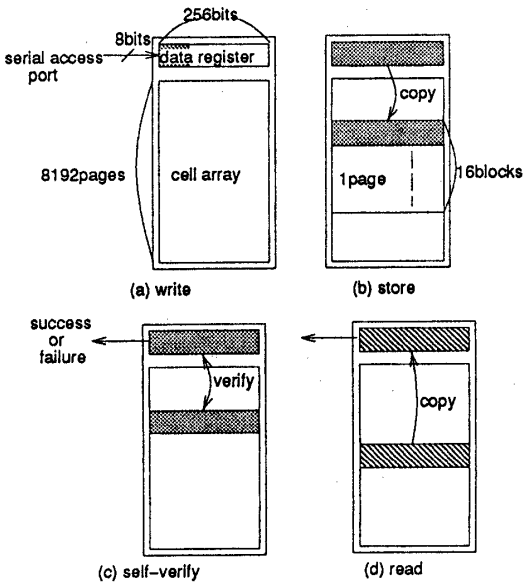

図 1: フラッシュメモリ

この方式ではアクセスが集中するページは短時間でディスクに書き込まれるが、そうでないページはディスクに書き込まれるまでの時間が非常に長くなる可能性がある。

不揮発メモリに障害が発生するとディスクに書き込まれなかったログは失われるので、回復処理はディスクに存在するログしか利用できない。特に、アクセスが集中していなかったページは検査点時刻以降のログがディスクに保存されていない可能性がある。この場合、データベースは検査点時刻までしか回復できない。

3. 本稿のログ管理方式

3.1 ログ保存

本方式では、以下のようにログを管理する。

- Step 1: ログを従来の方式と同様に生成順にフラッシュメモリへ保存する。
- Step 2: 逐次回復のためログ番号(LSN)を各ページごとに分類し不揮発メモリに保存する。

Step 1、2の終了後、データ更新をデータベースに反映する。以上の動作の概要を図2に示す。

3.2 回復処理

本稿のシステムは不揮発メモリに障害が発生した場合と発生しなかった場合の双方に対応する。以下にそれぞれの場合の処理について述べる。

[不揮発メモリに障害が発生しなかった場合]

各ページごとのLSNが不揮発メモリ中に存在するので、逐次回復を行いながらデータ処理を再開する。本稿のシステムで

Realization of Efficient Log Management and Recovery Utilizing Flash Memory
Hiroki TAKAKURA　Yahiko KAMBAYASHI
Faculty of Engineering, Kyoto University

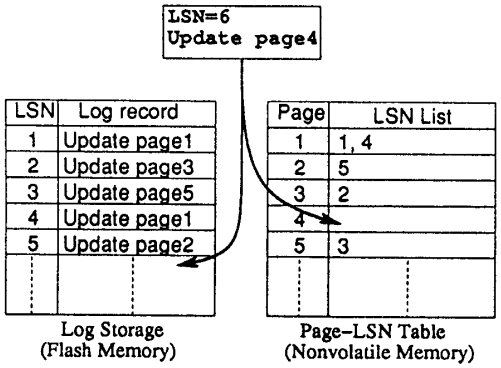

図 2: ログ管理の概要

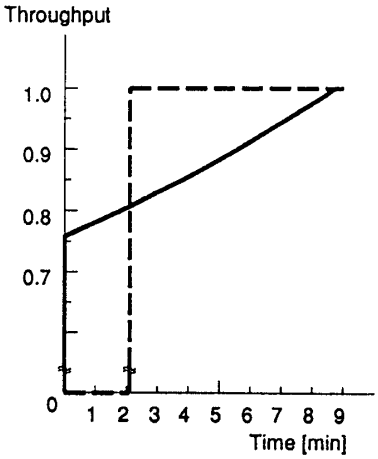

図 3: トランザクションスループット比

は、障害発生前にページにアクセスが集中していたか否かを判断するため各ページに更新回数を数える数ビットのカウンタを設ける。カウンタの値は更新が行なわれるたびに1つ増加するが、数えきれなくなったらオーバフローしたとしてそれ以上数えない。回復処理ではまずこのカウンタの値を読み、オーバフローしていればアクセスが集中していたページとして直ちに回復する。これらのページを回復した後にデータ処理を再開する。残りのアクセスが集中していなかったページはデータ処理と並行してアドレス順に回復される。但し、データ処理が未回復のページを必要とする場合はそのページを優先して回復する。回復処理は以下のように行なわれる。

- Step 1: 回復されるページのLSNを不揮発メモリから順次読み出す。
- Step 2: ページデータをバックアップされたデータベースから主記憶に転送する。
- Step 3: LSNが示すログをフラッシュメモリから読み出し、ページを再更新する。

[不揮発メモリに障害が発生した場合]

各ページを更新したログのLSNが失われているため逐次回復を実行できない。このため、データベース全体を回復した後にデータ処理を再開する。回復処理は以下のように行なわれる。

- Step 1: バックアップされたデータベースをアドレス順に主記憶に転送する。
- Step 2: 生成順(LSN順)にログをフラッシュメモリから読み出し、ページを再更新する。

4. 性能評価

本稿ではTPCベンチマークに基づいた解析を行なった。ここでは1000tpsのトランザクションスループットの場合について述べる。この場合、主記憶データベースのサイズは約10GBとなる。逐次回復では、アクセスが集中していたページの回復は数秒で終了するが、その後の集中していなかったページの回復には8分かかる[TAK93]。これに対してデータベース全体を一括して回復する場合は、2分で回復が終了する。

逐次回復では、アクセスが集中していたページの回復中は回復処理だけを行なうのでページの回復順序が前もって分かる。このため、フラッシュメモリ内のデータ転送を主記憶への転送に先行して行え、見かけ上メモリ内のデータ転送時間を0にできる。アクセスが集中していなかったページの回復中は、データ処理も並行して行なわれ、データ処理の要求する未回復ページが優先して回復されるため、ページの回復順序が予想しにくい。このため、メモリ内のデータ転送時間が回復時間に大きな影響を及ぼすことになる。これに対して一括回復の場合は、回復処理だけを行なうため、全てのページの回復順序が前もって分かり、メモリ内のデータ転送時間は回復時間に影響を及ぼさない。

しかし、逐次回復では1トランザクションあたりの待ち時間は最大で数msと一括回復の最大2分(システム停止時間)に比べ遥かに短い。これは、ディスクに比べ、フラッシュメモリで乱アクセスを行なう際のアクセス待ち時間が15μsと極めて短いためである。

通常時を1とした場合の回復処理経過時間に対するトランザクションスループット比を図3に示す。実線は逐次回復を行った場合の、破線は一括回復を行った場合のスループット比を表す。このように逐次回復では、一括回復に比べ、通常時の性能まで回復するのに時間がかかるが、性能の劣化は初期の段階でも20%程度である。

5. まとめ

今回は、フラッシュメモリをログ保存に利用することで、従来の不揮発メモリを利用する場合の問題点を解決し、さらに、場合に応じて逐次回復でも一括回復でも行なえる方式について述べた。また本研究では、バスを監視することにより主記憶に対する更新情報を読み出し、ハードウェア的にログを自動生成するシステムについて研究中である。ハードウェアログ管理システムによりシステムに影響をほとんど及ぼさないログ管理が可能になると考えられる。

参考文献

[KAM91] Y. Kambayashi, H. Takakura, "Realization of Continuously Backed-up RAMs for High-Speed Database Recovery," Database Systems for Advanced Applications '91, World Scientific, 1992, pp.236-242.

[LEH87] T.J. Lehman, M.J. Carey "A Recovery Algorithm for A High-Performance Memory-Resident Database System," Proc. ACM SIGMOD Conf., 1987, pp104-117.

[TAK93] H. Takakura, Y. Kambayashi, "Continuous Backup Systems Utilizing Flash Memory, " Int. Conf. on Data Engineering, 1993 (to appear).