M_046

多重書きと順序通知を用いた短応答時間データ冗長化方式

Short Response Time Data Redundancy Method by Multiple Data Writing and Order Notification

宮田美知太郎† 小川周吾† 長谷部賀洋†
Michitaro Miyata Shugo Ogawa Yoshihiro Hasebe

1. はじめに

近年、情報システムの急速な普及により電子データの重要性が増している。それに伴いデータの損失が発生した時の影響が増大している。このため、ストレージでは、システムの一部に障害が発生した場合でもデータが失われるリスクを低減する方法として、ミラーリングやレプリケーションのような手法が用いられる。これらの手法では、複数のデータ格納部に複製を保存する事によりデータを冗長化して、データ損失のリスクを抑える。

加えて、データウェアハウス等のように格納されたデータを複数の用途に使う目的や、無停止でバックアップを行う目的でも多数の複製が求められるようになってきている。

冗長化されたデータの複製は常に同一に保たれる必要がある。これらが同一に保たれない場合、一部の複製におけるデータ損失の発生を意味する。複製が常に同一に保たれるためには、データの一貫性維持、つまり全複製に対して同一順序でデータ更新を行う必要がある。

しかし冗長化を行う場合、一貫性維持の処理の追加に伴ってコマンド応答時間が増大する。そのため、冗長化と性能が両立しないという問題が発生する。

この問題に対して本稿では、多くの複製を作成した場合でも応答時間の短い冗長化方式を提案する。

2. 冗長化方式の比較

本章では、冗長化の基本方式および冗長化を正しく行うために必要な一貫性維持の説明、一貫性維持を実現する基本方式の比較を行う。

方式間の比較では特に Write コマンドの応答時間に寄与する通信回数に着目する。これは、CPU 等の性能向上によりストレージの格納部における処理時間は短くなり、応答時間に占める通信時間の割合が増加する傾向にあるためである。

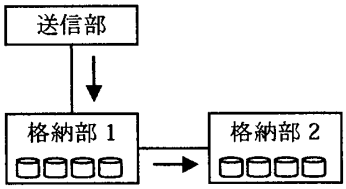
2.1 直列方式と並列方式

データを複数の格納部に保存する方式を大別すると、データを一旦一つの格納部で受けた後に他の格納部に転送する直列方式（図 1(a)）と、データを複数の格納部に一度に送る並列方式（図 1(b)）とがある。

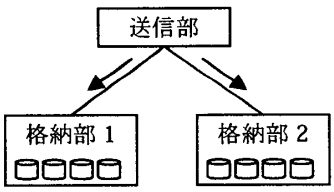
ここで、直列方式、並列方式の図は大きく分けて送信部、格納部から構成される。送信部は Write コマンドの発生源を持つ装置であり、格納部はあるアドレス空間のデータを格納する装置である。

直列方式にはさらに、格納部 1 が格納部 2 にデータを書き込んだ事を確認した後に送信部へ応答を返す同期式と、格納部 2 からの応答を待たずに応答を返す非同期式がある。同期式は応答時間が長くなるという欠点があり、非同期式は格納部 1 の障害が発生した時点で格納部 2 に転送されていなかったデータが失われるという欠点がある。

一方、並列方式は複数の格納部に並列にデータを送信する多重書きを行う方式である。並列方式は、データを一度に送るため、冗長度を上げた場合でも短い応答時間で Write を完了する事が可能である。このため、本稿では並列方式のみを検討の対象とする。



(a) 直列方式



(b) 並列方式

図 1 直列方式と並列方式

2.2 一貫性維持の要件

複製を作るにあたって、冗長化された複数の格納部間でデータの一貫性が維持される事が必要である。すなわち、特定の時点でいずれの格納部からデータを読み出しても同じデータが得られなければならない。このためには、複数の Write コマンドに対して全ての格納部で同じ順序でデータの格納が行われなければならない。

並列方式において、同一のアドレス領域に同時期に書き込みを行う送信部が複数有る場合、単純に各々の送信部が複数の格納部の同じアドレスにデータを並列に書き込む処理を行うと、データの一貫性が崩れる可能性がある。

例として図 2に示すように送信部 1,送信部 2 と冗長化された格納部 1,格納部 2 がネットワークで接続されているシステムを仮定する。このようなシステムでは送信部 1 と送信部 2 が送信するコマンドやデータが格納部 1 に到着する順序と格納部 2 に到着する順序は必ずしも一定ではない。

† 日本電気（株）システムプラットフォーム研究所
System Platforms Research Laboratories,
NEC Corporation

例えば、同時期に同じアドレスに対して送信部1がデータAを、送信部2がデータBを送信したとする。ネットワークの遅延等により格納部1にはデータA、データBの順で、格納部2にはデータB、データAの順でデータが到着する事が起こりうる。この時、先に到着して格納されたデータは後から到着したデータにより上書きされるため、格納部1の該当アドレスにはデータBが、格納部2の該当アドレスにはデータAが記録された状態となる。すなわち格納部1と格納部2でデータの一貫性が崩れた状態が発生する。

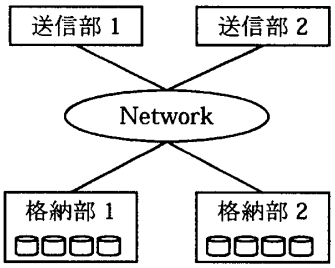


図2 複数の送信部を持つ例

2.3 一般的な一貫性維持方式の検討

前節で説明したような一貫性が崩れる状態を防ぐための方法として、最も単純な方法はコマンドを一箇所で整列する方法である（(1)コマンドの順序整列）。また、トランザクション処理で用いられるタイムスタンプや二相ロック［1］を応用した方法が考えられる（(2)コマンドへの順序付加、(3)二相ロック）。以下、それぞれの方法について述べる。

(1) コマンドの順序整列

Write コマンドを複数の格納部に送信する前に特定の一箇所で整列し、各々の格納部ではコマンドを受け取った順序に応じてデータを格納する方法である。全ての格納部には一箇所で整列された順序でコマンドが到着し、この順序でデータの格納が行われるため格納部間でデータの一貫性が維持される。

図2の構成において、送信部2でコマンドの順序整列を行う場合のWriteシーケンスの例を図3に示す。Writeコマンドを一旦送信部2に送り、送信部2でコマンドを整列する事により、格納部1と格納部2でコマンドが到着する順序が異なる状態が発生する事を防ぐ。

本方式は順序を整列する箇所がボトルネックとなりやすく、順序を整列する箇所へのコマンド転送が必要なため、応答時間が長くなるという欠点を持つ。

(2) コマンドへの順序付加

Write コマンドを複数の格納部に送信する前に送信元にてコマンドに順序を付加し、それぞれの格納部ではコマンドに付加された順序に従ってデータを格納する方法である。格納部によってコマンドが到着する順序が異なった場合でも、コマンドに付加された順序に従ってデータを格納するため、全ての格納部でのデータ格納順序は同じになり、一貫性が維持される。この方式では全ての送信部で整合性の取れた順序を付加する必要がある。

図2の構成において、送信部2が順序の決定を行う場合のWriteシーケンスの例を図4に示す。送信部1がWriteを行う場合、まず送信部2に順序問合せを行い、送信部2は付加する順序を送信部1に通知する。送信部1は送信部2より送られた順序をコマンドに付加して格納部1,2にそれぞれ送る。コマンドを受け取った格納部1,2はそれぞれ、付加された順序を見て、即座に処理してよいコマンドか、先行するコマンドが未着であるのかを判断する。先行するコマンドが未着である場合は、先行するコマンドの到着まで処理を保留する。

本方式では複数の送信元で一貫した順序を付加する必要があるため、送信元間での順序の問合わせが発生し、応答時間が長くなるという欠点を持つ。また、全ての格納部で順序に応じてコマンドを並びかえる処理が必要である。

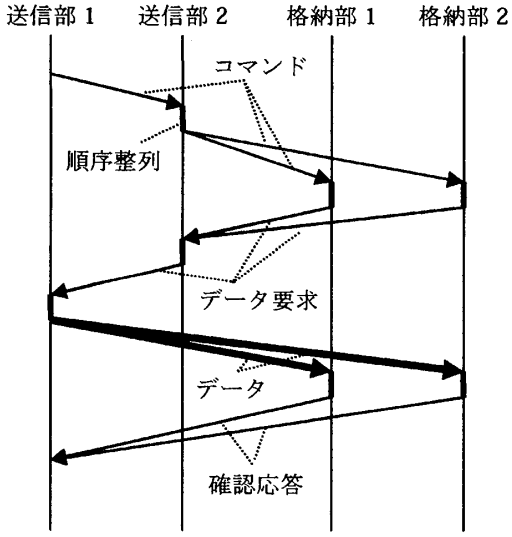


図3 Writeシーケンスの例
（コマンドの順序整列）

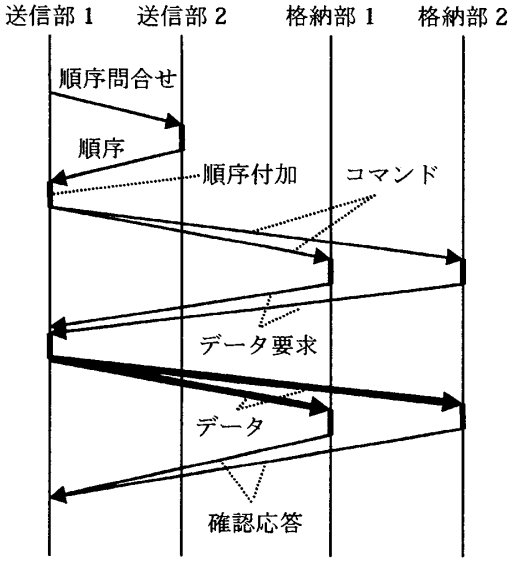


図4 Writeシーケンスの例
（コマンドへの順序付加）

(3) 二相ロック

送信部が特定の順序で格納部の記録領域のロックを行い、全てのロックが取れた時点でデータを書き込む方式である。すべての格納部がひとつの送信部によりロックされた状態でしかデータ書き込みが行われないため、すべての格納部でデータを格納する順序が一致し、一貫性が維持される。

図 2の構成において、本方式を用いた場合の Write シーケンスの例を図 5に示す。送信部 1はまず格納部 1にコマンドを送り、格納部 1では対象アドレス領域をロックする。格納部 1でロックが行われた事を確認すると、送信部 1は格納部 2にコマンドを送り、格納部 2では対象アドレス領域をロックする。格納部 2でもロックが行われると、送信部 1は格納部 1と格納部 2にデータを送信する。

本方式は、冗長数の増加に伴い通信回数が増えて応答時間が増加するという欠点を持つ。

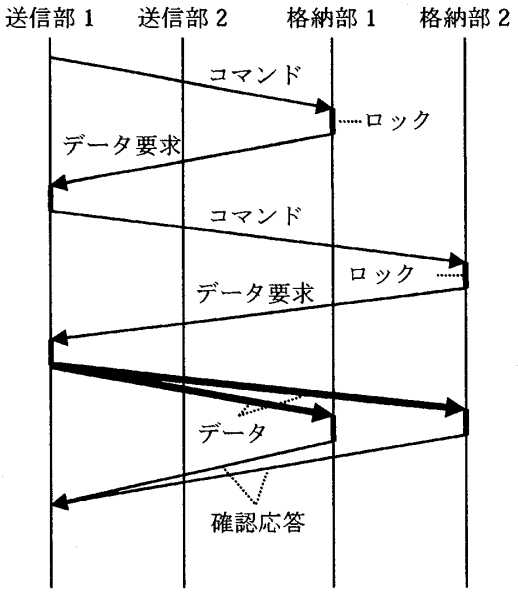


図 5 Write シーケンスの例（二相ロック）

3. 提案方式

本章では、我々の提案する、多重書きと順序通知を用いた冗長化方式について、ストレージの構成と一貫性維持の手順を説明する。また、提案方式と一般的な一貫性維持方式について、比較を行う。

3.1 冗長化方式の説明

本節では我々が提案する冗長化方式について説明する。提案する冗長化方式では、格納部のうち 1つをマスター、その他の格納部をレプリカとして区別する。また、レプリカとなる各格納部の前段に一時格納部を設ける。

図 6に本方式を用いた時の構成例を示す。本例は格納部 1をマスター、格納部 2をレプリカと位置づけた場合の 2冗長の構成である。冗長数を大きくする場合、マスターである格納部 1以外のすべての格納部の前段に同様に一時格納部を設ける。

次に、提案方式のシーケンスについて図 7を用いて説明する。図 7は図 6の構成において送信部 1が Write を行う場合のシーケンスの例である。

まず送信部 1は格納部 1と一時格納部に Write コマンドを送信する。マスターである格納部 1ではコマンドを受けるとデータを受信する領域を確保した後に格納部 1にデータ要求を返す。また、格納部 1はコマンドを受けた順序に従ってデータを保存する順序を決定し、コマンドと順序とを対応付ける情報をすべての一時格納部に通知する（図 7における「順序通知」）。一時格納部は送信部 1から送られたコマンドを受け取り、データを受信する領域を確保した後にデータ要求を送信部 1に送る。送信部 1からデータを受け取ると、データをコマンドと対応づけて一時的に保存する。さらに一時格納部は格納部 1から受信した順序情報に従って、格納部 2にデータを格納する順序を決定する。以上の手順で決定された順序に従ってレプリカである格納部 2にデータを転送する。

この動作により、すべての格納部においてマスターにおけるデータ保存の順序と同じ順序でデータが保存される。このため格納部間でデータの一貫性が崩れる事が無い。また、一貫性を維持するために行われる順序通知は他の通信と並行して行われるため、応答時間に影響しない。このため、応答時間は冗長数に拠らず冗長の無い場合（格納部が 1つだけの場合）と比べて増加しない。

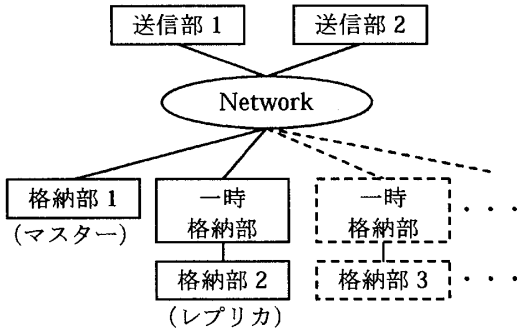


図 6 提案方式の構成例

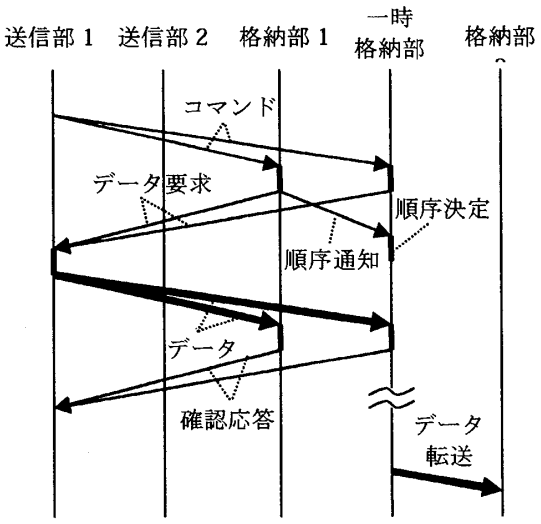


図 7 Write シーケンスの例（提案方式）

表1 冗長化方式の比較

	提案方式	コマンドの順序整列	コマンドへの順序付加	二相ロック
通信回数	4	6	6	2n+2
格納部の区別	有	無	無	有
順序決定箇所	マスター	送信部	送信部	格納部
順序整列箇所	一時格納部	送信部	格納部	送信部

・通信回数は、応答時間に寄与するもののみ。並列に処理される通信は除外している。
・nは複製の数（冗長数－1）である

なお、データのReadは、マスターである格納部1のみから行う。

次に、図8を用いて具体的に格納部1と格納部2の前段の一時格納部に到着するWriteコマンドの順序が異なった場合でも格納部1と格納部2では同じ順序でデータが格納される事を説明する。

送信部1がWriteコマンドA（以降、コマンドA）を、送信部2がWriteコマンドB（以降、コマンドB）をほぼ同時に送信部1と一時格納部に送信し、格納部1にはコマンドA,コマンドBの順で、一時格納部にはコマンドB,コマンドAの順でコマンドが到着したものとする。一時記憶部はコマンドが到着した順、すなわちコマンドB,コマンドAの順でコマンドおよびデータを記憶する。一方、格納部1はコマンドA,コマンドBの順でコマンドを処理し、コマンドAを順序1,コマンドBを順序2として順序情報を一時格納部に通知する。一時記憶部は格納部1から受け取った順序情報をコマンドと対応して記憶する。図8はこの時の一時記憶部の状態である。この後、一時記憶部は順序に従ってコマンドA,コマンドBの順で対応するデータを格納部2に転送する。このため、格納部1と格納部2では、いずれもデータA,データBの順でデータが格納され、一貫性が維持される。

順序	コマンド	データ
2	コマンドB	データB
1	コマンドA	データA
・・・	・・・・	・・・・・

図8 一時記憶部におけるデータの例

3.2 他方式との比較

表1に提案方式と、2.3で説明した3つの方式との比較を示す。提案方式は他の方式に比べて応答時間の増加につながる通信回数が最も少ない。また、提案方式における通信回数は冗長数に依存しない。このため複製間のデータ一貫性を維持しながら短応答時間が必要なストレージシステムにおける冗長化方式として適している。

4. 基礎検証実験

提案方式の基本動作を確認するため、プロトタイプによる検証を行った。送信部、格納部、一時格納部の各機能をLinux上で動作するソフトウェアとして実装し、IAサーバ上で動作させた。サーバ間は1Gb Ethernetにて接続した。

動作検証の結果、マスターである格納部と一時格納部の間でWriteコマンドの到達順序が異なる場合でも格納部間でデータの一貫性が維持される事を確認した。

5. おわりに

本稿では、データの冗長化を行う場合に一貫性を維持するためにコマンド応答時間が増大する問題について述べた。我々は、多くの複製を作成した場合でも一貫性を維持でき応答時間の短い冗長化方式として、多重書きと順序通知を用いた方式を提案した。提案方式は他の一般的な一貫性維持方式に対して、コマンド応答時間に寄与する通信回数が少なく、高性能化が望める事を説明した。また、プロトタイプによる検証により一貫性維持が出来る事を確認した。

参考文献

[1] フィルマン,フリードマン：協調型計算システム ー分散型ソフトウェアの技法と道具立てー 初版, マグロウヒルブック, 1986