ログベースディザスタリカバリシステムにおけるログ適用高速化方式の評価

Evaluation of Speed-up Methods of Database Redo Processing for Log-based Disaster Recovery Systems

渡辺 聡♠ 鈴木 芳生♦
水野 和彦♦ 藤原 真二♠

Satoru WATANABE Yoshio SUZUKI
Kazuhiko MIZUNO Shinji FUJIWARA

広域災害対策として遠隔地にデータをバックアップするディザスタリカバリシステムでは，メインサイトが被災した場合に，リモートサイトを代替システムとして使用する．しかし，代替システムとしての用途だけでは，平常時にリモートサイトが有効に活用されない問題がある．そこで，平常時から，リモートサイト側でデータ集計処理などを実行することが考えられる．しかし，従来，リモートサイトをバックアップ以外の目的に使用すると，リモートサイト側で実行するログ適用処理に遅延が生じ，データのバックアップに障害が生じる可能性があった．本研究では，ストレージ装置に対するIO要求の並列化，および，IOのシーケンシャル化技術を用いてログ適用処理を高速化した．これらの技術による性能向上率を明らかにするために，4つの異なる実装方式で評価を行ない，ログ適用処理の性能を5.2倍に向上できることを示した．本技術を用いることにより，リモートサイトの有効活用が可能になる．

For high availability of computer systems, online remote backup systems called disaster recovery systems are becoming common mainly in enterprise mission-critical systems. In disaster recovery systems, the remote site is used as the substitute system when the main site collapses due to hurricanes, earthquakes, and so on. The remote site is seldom used, and there is wastage of the computer resources. The purpose of this paper is to enable effective use of the remote site in normal times. For this purpose, it is needed to speed up the database redo processing in log-based disaster recovery systems. We used parallelization technique and sorting log technique to speed up the redo processing. We developed four prototypes by different implementation methods and speeded up the redo processing by 5.2 times. These techniques enable effective use of the remote site.

♠正会員 株式会社日立製作所 中央研究所
{satoru.watanabe.aw, shinji.fujiwara.yc}@hitachi.com
♦株式会社日立製作所 中央研究所
{yohiso.suzuki.rf, kazuhiko.mizuno.pq}@hitachi.com

1. はじめに

企業活動において情報システムの果たす役割が増大し，情報システムのリスク管理が企業の重要な課題になっている．リスク管理を怠る代償は大きく，災害などで情報システムが停止した場合，企業は甚大な損失を被る可能性がある[1]．そのため，災害に備えてメインサイトの遠隔地に設置したリモートサイトにデータをバックアップするディザスタリカバリ（DR）システムが注目されている[2][3]．

現在，企業の情報管理においてデータベースが重要な役割を果たしている．そのため，データベースのデータ保護は特に重要な課題である．データベースのDRシステムでは，ログベースのDR方式が広く用いられ，商用データベースにおいても製品化されている[4]．ログベースのDR方式では，データ更新ログをメインサイトからリモートサイトにコピーし，データ更新ログの内容をデータに反映するログ適用処理をリモートサイト側で実行することで，データをバックアップする．

従来，リモートサイトは，災害などでメインサイトに障害が生じた場合の代替システムとして使用されていた．しかし，代替システムとしての用途だけでは，平常時にシステムリソースが有効に活用されない問題がある．そこで，リモートサイト側でデータ集計などの処理を実行することで，平常時からシステムリソースを有効に活用することが考えられる．

しかしながら，ログ適用処理と並行してリモートサイトを使用すると，処理能力が不足し，ログ適用処理の遅延によるシステム障害が生じる可能性がある．そこで，ログ適用処理の遅延回避を目的として，ログ適用処理の高速化に向けた技術開発を行った．例えば，ログ適用処理を2倍に高速化できれば，夜間にログ適用処理を実行し，残りの時間にリモートサイトの活用が可能になる．本研究では，ログ適用処理の高速化方式を検討し，高速化方式を実装した場合の性能向上率の評価を行った．

2. 関連研究

データベースが管理するデータを遠隔地にバックアップする方式として，Dual Input方式やRemote Mirrored Disk方式などがある[5]．[5]において，導入の容易性や性能などの観点から方式の比較が行われ，本研究が対象とするログベースのDR方式が有力と結論されている．[6]において，ログベースのDR方式の実験システムが開発され，実装方式の違いによる性能特性が評価されている．

ログ適用処理の改良に関しては，[7]において，低オーバヘッドでログ適用処理を並列化する技術が提案されている．[8]では，ログ並べ替え技術によりストレージに対するIOをシーケンシャル化し，ログ適用処理を高速化している．[8]では，ソフトウェアRAIDを用いた評価結果が示されている．我々は，ハードウェアRAIDの環境において，並列化とログ並べ替えの両方の技術を4つの方式で実装し，評価を行った．

3. ログ適用高速化技術

3.1 高速化の基本方針

ログ適用処理にタイムスタンプを挿入し，日立製サーバとSANRISE 9570Vを用いたシステム上で動作させ，ログ適用処理の処理時間の内訳を調べた．データ，および，トランザクションは5章に示すものを使用した．図1に示すように，ログベースのDRシステムでは，メインサイトのデータベー

スが出力するデータ更新ログとデータ本体のうち，データ更新ログだけをリモートサイトに転送し，リモートサイト側でログ適用処理を行う．ログ適用処理は，ログ読込みステップ，ログ解析ステップ，および，データ更新ステップから構成され，処理時間の内訳はそれぞれ 1%，4%，95%であった．

ログ読込みステップはログファイルからログバッファにログを読込むステップであり，ログを順に読込むシーケンシャルアクセスであることから処理時間の割合は小さい．また，ログ解析ステップは，ログ適用に必要なログだけを抽出するステップであり，DB ヘッド（ログ適用を行なう専用装置）のメモリ上の操作であることから，この処理時間の割合も小さい．処理時間の割合が全体の 95%を占めるデータ更新ステップは，さらに次の 3 つの処理に分けられる．

1. 更新対象のデータ領域（ページ）をストレージから DB バッファに読込む処理
2. ログの内容をページに反映する処理
3. ログの内容を反映したページをストレージに書き戻す処理

2 番目のログの内容をページに反映する処理は DB ヘッドのメモリ上の操作であり，処理時間の割合はログ適用ステップの約 2%で小さい．1 番目の処理と 3 番目の処理において，ストレージに対するデータ入出力の処理時間が大きいことが，データ更新ステップの処理時間が大きい要因である．

さらに，データ入出力の Read と Write の処理時間の割合を IO トレースにより調べた．その結果，Read の処理時間が 92%を占めていた．これは，Write 処理では，ストレージキャッシュにデータを書込み即座に応答を返すキャッシュ Write 機構が働くのに対し，Read 処理では，（キャッシュミスの場合には，）ハードディスクドライブからデータを読込む必要があるためである．そこで，本研究では，データ更新ステップのデータ入出力のうち，主にストレージから DB バッファにページを読込む処理の高速化技術を検討した．

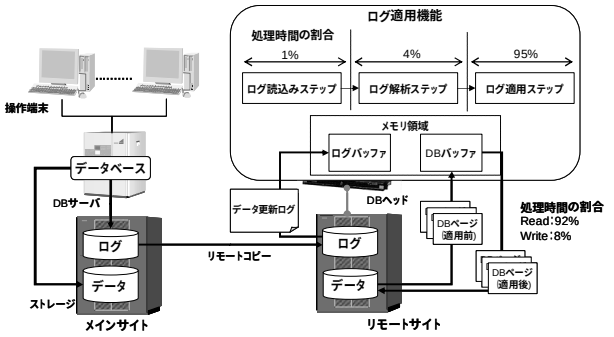

図 1　ログ適用処理の処理フロー

Fig.1　Processing flow of database redo

3.2 高速化技術の検討

ログ適用ステップの高速化技術として，図 2 に示すように，(1)ログ並べ替え，(2)ログ適用プロセスの並列化の，2 つの技術を検討した．

高速化技術(1)：ログ並べ替え

ログ並べ替えステップを設け，ログバッファに格納されているログをページ番号順にソートする．ページ番号は，ロジカルブロックアドレス（LBA）の順に付与されることから，ページ番号による並べ替えにより，ストレージに対する IO がシーケンシャルになる．ストレージの性能はランダムアクセスよりシーケンシャルアクセスの方が高いことから，ログ並べ替えにより高速化が可能である．なお，並べ替えによりログの適用順序が変更されるが，今回評価に用いた日立製データベース HiRDB では，同一ページに対するログの適用順序を守れば正常にデータを回復でき，異なるページのログは順序を変更して適用できる．

高速化技術(2)：ログ適用プロセスの並列化

DB バッファにページを読み込み，ログの内容を反映するログ適用プロセスを複数個起動し，データの入力を並列に実行する．ストレージ装置は並列して IO を処理できるため，ログ適用プロセスの並列化により高速化が可能である．

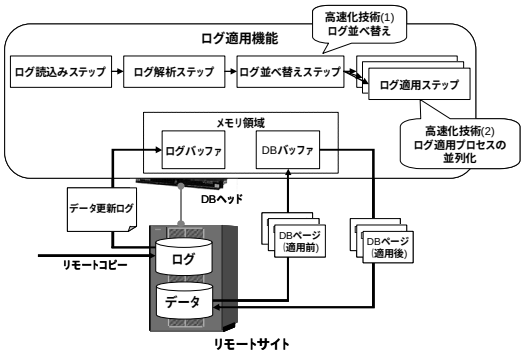

図 2　ログ適用処理の高速化技術

Fig.2 Speed-up techniques of database redo

4. プロトタイプの実装方式

以下に示す 4 つの方式で，プロトタイプを実装した．各方式の詳細は以下の通りである．

1. LBA レベルスケジューリング

ログをページ番号順にソートし，ストレージの LBA（ロジカル・ブロック・アドレス）順にシーケンシャルな IO を発行する．ソートしたログをログ適用プロセスに振分ける方法で以下の 3 つのバリエーションがある．

(i) Single Stream 方式
- ログ適用プロセスを 1 個だけ起動し，全てのログを 1 つのプロセスに振分ける
- データ領域に対して 1 ストリームのスパースなシーケンシャル IO が発行される

(ii) Partitioned Multi Stream 方式
- ログ適用プロセスを複数個起動し，ソートしたログをプロセスの個数に等分して振分ける
- データ領域に対して，プロセスの個数分のスパースなシーケンシャル IO が発行される

(iii) Unified Multi Stream 方式
- ログ適用プロセスを複数個起動し，IO が完了するごとに 1 つずつログをログ適用プロセスに振分ける
- データ領域に対して，1 ストリームのスパースなシーケンシャル IO が発行される

2. 物理レベルスケジューリング

RAID ディスクでは，複数のドライブに分散してデータが格納される．LBA レベルスケジューリングでは，IO 要求がドライブで競合する可能性がある．IO 要求の競合を回避するため，物理レベルスケジューリングでは，ドライブ数分のログ適用プロセスを起動し，ログ適用プロセスとドライブを対応付ける．RAID のストライピング方式を事前に調べておき，ソートしたログを，ドライブごとに分類してログ適用プロセスに振分ける．物理レベルスケジューリングでは，ドライブごとにスパースなシーケンシャル IO が発行される．

5. 評価

5.1 評価環境

図 3 に示す評価環境を用いてログ適用高速化方式の評価を行なった．メインサイトは DB サーバとストレージから構成され，サーバとストレージはファイバチャネル（FC）スイッチを介して接続されている．リモートサイトでも DB サーバとストレージが FC スイッチを介して接続されている．メインサイトとリモートサイトのストレージは転送距離を拡大するためのチャネルエクステンダを介して接続されている．リモートサイトの FC スイッチには FC アナライザが接続され，これによりストレージに対する IO のトレースを取得できる．表 1 に示すように，DB サーバと DB ヘッドには日立 H9000V rp2400 を使用し，ストレージには SANRISE9570V を使用した．

表 2 に示すように, メインサイトの DB サーバではデータベース（HiRDB）が動作し，データベースに負荷を与えるトランザクション実行ツールが動作する．トランザクション実行ツールはデータベースシステムの標準ベンチマークである TPC-C[9]を模して作成されたツールである．データを格納したロジカルユニットの容量は 35GB であり, 実際のデータの容量は約 30GB である．また，DB ページのサイズは 4KB とし，データは 3D+1P の 4 個のドライブで構成される RAID5 のロジカルユニットに格納した．

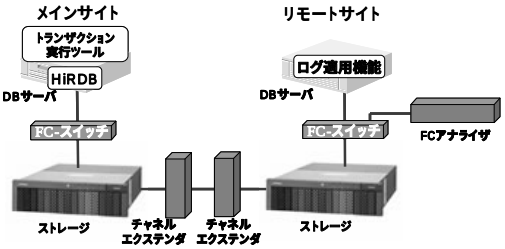

図 3 評価環境
Fig.3 Experimental environment

表 1 ハードウェア構成
Table 1 Hardware configuration

DB サーバ	H9000V rp2400 , 650MHz, 2Way, メモリ 1GB
ストレージ	SANRISE9570V キャッシュ 1GB 3D+1P 構成の RAID5 の LU にデータを格納（容量 35GB）

表 2 ソフトウェア構成
Table 2 Software configuration

データベース	HiRDB SingleServer V7.0
トランザクション実行ツール	TPC-C を模したツール Warehouse 数：250，データサイズ：30GB.トランザクション比率：neworder:25%,payment:13%,orderstatus:58%,delivery:2%,stocklevel:2%
ログ適用機能	ログ適用機能のプロトタイプ

5.2 評価方法

4 章に示した方式で実装したプロトタイプのうち，並列化技術を用いる方式では，ログ適用のプロセス数がパラメータになる．但し，物理レベルスケジューリングのログ適用プロセス数は，データを格納しているドライブの個数により決定され，今回の評価環境におけるログ適用プロセス数は 4 である．そこで，物理レベルスケジューリングと LBA スケジューリングの効果を比較するため，表 3 に示すように，並列化技術を用いる方式では，ログ適用プロセス数 4 と 10 で評価を行った．

ログ並べ替え技術では，並べ替えるログの容量（ソート領域容量）がパラメータになる．これを変更した場合の効果を評価するため，表 3 に示すように，0MB（並べ替えなし）から 256MB までの範囲で評価を行った．

表 3 評価パラメータ
Table 3 Parameters of experiment

実装方式 ＼ 高速化技術	LBA レベルスケジューリング			物理レベルスケジューリング
	Single Stream 方式	Partitioned Multi Stream 方式	Unified Multi Stream 方式	
並列化	×	○	○	○
ログ適用プロセス数	1	4,10	4,10	4
ソート領域容量	0MB（並べ替えなし），64MB，128MB，256MB			

ログ適用処理のスループットを測定するため，メインサイトで 512MB のログを生成し，リモートサイトにログをコピーした後に，ログ適用機能を起動して，512MB のログの適用に要した時間を測定した．なお，ログのコピーとログ適用を並列して実行した場合にも，ログ適用の性能は同等になることを確認している．

5.3 評価結果

評価に用いた IO のパターンを説明するため，Single Stream 方式で 256MB のログを並べ替えてログ適用した場合の，データアクセスの IO トレースを図 4 に示す．図 4 に示すように，35GB の領域に格納された約 30GB のデータに対して，シーケンシャルなアクセスが行われる．ストレージから DB バッファにページが Read され，ログの内容が反映されたページは，DB バッファが満杯になったタイミングで Write される. DB バッファは, LRU 方式で管理されており，Write は Read に遅れて実行される．なお, 3.1 節で述べたように，ストレージキャッシュにデータを書き込んで即座に応答を返すキャッシュ Write 機構が働くため，Write は高速に実行される．そのため，ログ適用処理の性能に大きく影響するのは, DB バッファにページを読み込む Read 処理である．

256MB のログには, ログレコードが約 68 万個含まれていた．各ログレコードには更新対象のページ番号と更新内容が記載されている．同じページを更新対象にするログレコードがあるため，更新対象のページの総数は約 13.5 万ページであった．1 ページは 4KB であるので，更新対象のページの合計容量は約 540MB であり，これはデータ容量 30GB の 1.8%である. 図 4 では, IO が連なっているように見えるが, 実際はスパースなシーケンシャル IO が発行されている．

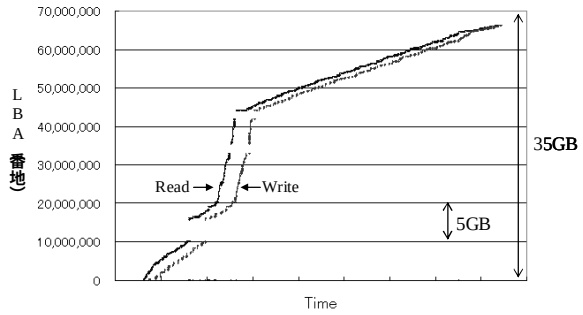

図 4 データアクセスの IO トレース
Fig.4 IO trace of database access

ログ適用処理のスループット性能を図 5 に示す. 図 5 の縦軸は Single Stream 方式で並べ替えなし場合の性能を 1 とした性能の倍率，横軸はソートするログの容量である．

ソートするログの容量が増えるほど IO がランダムアクセスからシーケンシャルアクセスに近づき，性能が向上する．Unified Multi Stream 方式(10Process)の性能が最も高く，最大 5.2 倍に性能向上した．また，ログ適用プロセス数が増加するほど, IO の並列度が上がり, 性能が向上する. Unified Multi Stream 方式（10Process）の並べ替えなしの性能は，Single Stream 方式の並べ替えなしの性能の 2 倍であった．

Partitioned Multi Stream 方式の性能は, 図 5 の点線で示され，Single Stream 方式の性能より低い．これは，Partitioned Multi Stream 方式でログ適用をすると，全体として IO がシーケンシャルにならないため，並べ替えの効果を打ち消してしまうためと考えられる．

物理レベルスケジューリングの性能は，Unified Multi Stream 方式（4Process）と同等であり，図 5 では、2 本のグラフがほぼ重なっている．このように，本評価では，ドライブ構成を考慮した物理レベルスケジューリングの効果は得られなかった. これは, データアクセスに偏りがないため, Unified Multi Stream 方式でもドライブに対するアクセスが分散し，物理レベルスケジューリングと同等の効果が得られたためと考えられる．

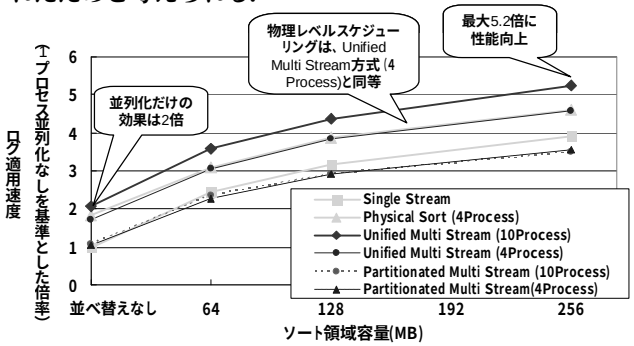

図 5　ログ適用のスループット性能

Fig.5　Throughput of database redo processing

メインサイト側では，大量の IO を並べ替えて IO を発行することはできない．一方，リモートサイト側では，IO の並べ替えが可能であり，ログ適用を高速に実行できる．Unified Multi Stream 方式を用いることで, ログ適用の性能がメインサイトの性能を上回り，リモートサイトの有効活用が可能になる．なお，評価環境のメインサイトでは，データの参照と更新の両方が実行されており，その性能は，図 5 の倍率 1 の性能と同等であった．本環境では，Unified Multi Stream 方式により，メインサイトの 5.2 倍の性能でログ適用できた．

6. まとめ

企業情報システムのリスク管理のニーズ増大を受け, 遠隔地にデータをバックアップするディザスタリカバリシステムが注目されている．ディザスタリカバリシステムでは，メインサイトが被災した場合にリモートサイトを代替システムとして使用する．しかし，代替システムとしての用途だけでは，システムリソースが有効に活用されない問題がある．そこで，リモートサイト側でデータ集計処理などを実行することで，システムリソースを活用することが考えられる．これを実現するため，ログ適用処理の高速化に向けた技術開発を行った．ストレージ装置に対する IO 要求の並列化，および，IO のシーケンシャル化技術を 4 つの方式で実装した．性能評価の結果，Unified Multi Stream 方式により，ログ適用処理の性能を 5.2 倍に向上できた．本技術を用いることにより，リモートサイトの有効活用が可能になる．

[謝辞]

本研究の一部は，文部科学省リーディングプロジェクト e-Society 基盤ソフトウェアの総合開発「先進的なストレージ技術」で技術開発された成果が反映されています．研究の推進にあたり東京大学生産技術研究所喜連川教授にご助言いただきました．感謝いたします．

[文献]

[1]Steven R. Christiansen, Lawrence L. Schkade, “Financial And Functional Impacts Of Computer Outages On Business”, University of Texas working paper #CRIS-87-01 (1987).

[2]”Interagency Paper on Sound Practices to Strengthen the Resilience of the U.S. Financial System”, 米国証券取引委員会(SEC)勧告 (2003.4.7).

[3]経済産業省 企業における情報セキュリティガバナンスのあり方に関する研究会報告書(平成 17 年).

[4]日立製作所,スケーラブルデータベース HiRDB Version 8 製品パンフレット(No:CA-567),(2006.6).

[5]D.L.Burkes, R.K.Treiber, “Design Approaches for Real-Time Transaction Processing Remote Site Recovery”, Compcon Spring '90. 'Intellectual Leverage'. Digest of Papers. Thirty-Fifth IEEE Computer Society International Conference, pp. 568-572, Feb. 1990.

[6]Christos A. Polyzois, Hector Garcia-Molina, “Evaluation of Remote Backup Algorithms for Transaction- Processing Systems”, ACM Transactions on Database Systems, Vol.19, No.3, pp. 423-449, Sep. 1994.

[7]Mohan C., Treiber K., Obermarck R., “Algorithms for Management of Remote Backup Data Base for Disaster Recovery”, Data Engineering, 1993. Proceedings. Ninth International Conference, IEEE, pp.511-518, Apr. 1993.

[8]合田和生,喜連川優,”データベース再編成機構を有するストレージシステム”,情報処理学会論文誌データベース, Vol.46 No.SIG 8(TOD 26), pp.130-147(2005.6).

[9]Transaction Processing Performance Council, “TPC Benchmark C, Standard Specification, Revision 5.1”, December 2002.

渡辺 聡　Satoru WATANABE

1999 年早稲田大学理工学研究科修士課程修了．同年，日立製作所入社．中央研究所研究員．日本データベース学会員．

鈴木 芳生　Yoshio SUZUKI

1994 年筑波大学大学院理工学研究科修士課程修了．同年, 日立製作所入社. 中央研究所主任研究員. 情報処理学会会員.

水野 和彦　Kazuhiko MIZUNO

1992 年熊本県立熊本工業高等学校電子科卒．同年，日立製作所入社．中央研究所研究員.

藤原 真二　Shinji FUJIWARA

1990 年京都大学大学院工学研究科情報工学専攻修士課程修了．同年，日立製作所入社．中央研究所主任研究員．1998～1999 年米国スタンフォード大客員研究員. IEEE, ACM, 情報処理学会, 情報通信学会, 日本データベース学会各会員.