

同期リモートコピーを用いたデータ欠損のない DR システムの性能安定性の評価

鈴木 芳生^{†1} 渡辺 聡^{†1} 水野 和彦^{†1}
藤原 真二^{†1} 河村 信男^{†2} 喜連川 優^{†3}

本論文では、同期リモートコピーを用いたデータ欠損のない DR システムを対象として、性能安定性の評価を行った。最初に、転送遅延がログや DB の出力に及ぼす影響を考慮して、正サイトで稼動する DBMS の性能モデルを構築した。回線シミュレータを用いた評価実験を行ったところ、性能モデルによって約 1 割の誤差で DR システムの性能を予測可能なことを確かめた。この性能モデルからログのみを同期リモートコピーで転送する方式は、ログ・DB とも転送する方式に比べて性能安定性が高いと考察された。そのため、同環境において回線遅延/帯域幅/入力負荷を変動させた場合の DR システムの性能を評価した。その結果、ログのみを同期する DR システムは、ログ・DB とも転送する DR システムに比べて、転送遅延や入力負荷といった環境の急激な変動に対して頑強であり、高い性能安定性を示すことが確かめられた。

Evaluation of Performance Stability of No Data Loss Disaster Recovery Systems Using Synchronous Remote Copy

YOSHIO SUZUKI,^{†1} SATORU WATANABE,^{†1}
KAZUHIKO MIZUNO,^{†1} SHINJI FUJIWARA,^{†1}
NOBUO KAWAMURA^{†2} and MASARU KITSUREGAWA^{†3}

For high availability of computer systems, online remote backup systems called disaster recovery (DR) systems are becoming common mainly in enterprise mission-critical systems. In this paper, we evaluated performance stability of No Data Loss DR system using synchronous remote copy. First, we build a performance model of DR system to evaluate performance stability. We confirmed that this model can predict throughput of DR system at error of about 10% in certain circumstances by comparing to experimental results using network simulator. From this model, we speculate that log-based DR

system using synchronous remote copy is more stable than storage-based DR system using synchronous remote copy. And we evaluated performance stability in experimental environment in which environmental parameter like circuit delay fluctuates. As a result of experiment, we confirmed that log-based DR system is robust over environmental fluctuation and more stable compared to storage-based DR system using synchronous remote copy.

1. 背景と目的

9.11 テロ以降、災害であってもビジネス（業務）を継続させることの重要性が再認識されている。ビジネス継続ができないことによる機会損失は大きく、たとえば、金融業界では時間あたり数百万ドルもの損失が生じるといわれている¹⁾。また、回復に長期間を要する場合、あるいは、データに大きな損失があった場合は、たとえビジネスを再開できたとしても、そのような企業の 51% は 2 年以内に廃業に追い込まれるなどと報告されている^{2),3)}。そのため、米国では社会全体への影響が多い金融業界に関しては、法規制を設ける動きもある⁴⁾。

同様に、日本でも経済産業省によってビジネス継続のためのガイドラインが作成されている⁵⁾。さらに、国際的にも英国 BS25999⁶⁾ (旧 PAS56⁷⁾) ならびに米国 NFPA1600⁸⁾ を土台として 2010 年までに ISO における標準化が予定されている。このような背景から、システムを二重化し、平常時から正サイトのデータを遠隔地に設置した副サイトにコピーしておき、災害時には副サイトでビジネスを継続するディザスタリカバリ (DR) 技術が注目を集めている。DR は従来からある技術だが、近年は、地震や大規模停電など広域災害であってもビジネスを継続させること、すなわち、副サイトをより遠隔地に設置することへの要求が高まっている。

日本では地震など広域災害への対応が必要である。そのため、都市間 DR、すなわち、サイト間の距離が数百 km 以上離れた構成での運用が必要となる。金融業界などデータ欠損が大きな損失を及ぼす場合には、都市間 DR においても、災害時にデータ欠損がないことが大きな価値となる。しかし、同時に、オンライン性能の確保、すなわち、副サイトへのコ

^{†1} 株式会社日立製作所中央研究所
Central Research Laboratory, Hitachi, Ltd.

^{†2} 株式会社日立製作所ソフトウェア事業部
Software Division, Hitachi, Ltd.

^{†3} 東京大学生産技術研究所
Institute of Industrial Science, the University of Tokyo

ピー処理が正サイトで行うオンライン処理に与える影響を最小限にしなければならないという課題がある。

さらに、実際に DR システムを運用していくためには、日々の運用が容易に実施できる必要がある。すなわち、性能設計や性能チューニングが容易に行える必要がある。そのためには、DR システムが転送遅延や AP (アプリケーション) 入力負荷といった環境因子の変動に対して高い性能安定性を有する必要がある。そこで、本論文では、災害時にもデータ欠損なくデータを回復することが可能な DR システムとして、同期リモートコピーを用いた DR システムの性能モデルを構築し、副サイトへのコピー処理が DBMS の内部挙動や性能に与えるメカニズムを明らかにする。さらに、回線シミュレータを用いた DR 評価環境を用いて、回線遅延/帯域幅/入力負荷を変動させた場合の DR システムの性能安定性を評価する。

本論文の構成は以下のとおりである。まず、2 章で関連研究について述べた後、3 章で代表的な DR システムの特徴と提案する DR システム、および、評価の対象とした DR システムについて述べる。4 章では、対象とした DR システムの性能モデルを構築するとともに、そのモデルをもとに性能安定性を考察する。5 章では、回線シミュレータを用いた評価実験により性能モデルの検証を行うとともに回線遅延など環境パラメータに変動を与えたときの DR システムの性能安定性を評価し、最後に 6 章において結論を述べる。

2. 関連研究

DBMS の DR システムとしては、サーバ間転送を用いたログベースの研究が多く行われている。ログベースの方式としては、副サイトへの適用と同期して正サイトのコミットが完了する 2-safe 方式、および、正サイトでのコミットは、副サイトへのコピー処理とは非同期で行われる 1-safe 方式が知られている^{9)–12)}。ただし、2-safe 方式では、2-phase commit に従って副 DB への適用までを正サイトのトランザクションに含める場合と、ログだけを副サイトに同期で送っておき、適用は非同期で行う場合がある。Polyzois らの研究⁹⁾では、マルチサーバでの分散トランザクション環境において、トランザクション整合性や正副サイトのサーバ間、あるいは、副サイト内の複数のサーバの間で、トランザクションの調停のために行われる通信量などを考慮してログ転送からログ適用までを効率的に実施する方式の提案・評価を行っている。

近年では、多くのシステムでリモートコピー機能を有するストレージを用いて DR システムが構築されている^{13)–15)}。リモートコピー機能を有するストレージを用いる利点としては、ストレージ層でデータ転送が保障されるため上位サーバで動作するアプリケーション

(AP) への転送処理の組み込みが不要であり、システムが単純化されることから、運用が容易であるという点があげられる。さらに、リモートコピーを利用した場合には、転送にサーバリソースを使用しないため、AP の性能確保が可能といったメリットもある。

また、システムの堅牢性に着目した研究も行われている。たとえば、文献 16), 17) では、災害時にもデータアクセスを継続的に行えるように、分散システムにおけるサーバ配置、レプリケーション方法、障害検出方法の検討を行っている。

本研究では、ストレージのリモートコピーを利用した DBMS の DR システムを対象として、転送遅延が DBMS 内部挙動に与える影響を考慮した性能モデルを構築するとともに、回線遅延や AP 入力負荷といった環境変動が生じた場合のシステムの性能安定性評価を行った。これまでの DBMS の DR システム性能評価に関する研究では、サーバ間転送のログベース DR システムの 1-safe と 2-safe の性能比較、あるいは、マルチサーバでの分散トランザクション環境下での、ログベース DR システムのサーバ間転送を削減する方式の提案・評価などが行われている⁹⁾、ストレージのリモートコピーを利用した DBMS の DR システムの性能安定性という観点で行った研究は、著者らが知る限りほかにはない。

3. 広域災害向け DR システム

本章では、地震などの広域災害に対応した DR システムについて説明する。

3.1 ストレージ主体の DR システム

近年、リモートコピー機能^{13), 27)}を有する高機能ストレージを用いた DR システムが多く構築されている^{13)–15)}。リモートコピー機能は、ネットワークを介して接続された正副ストレージ間で動作する機能であり、正ストレージへ発行された write I/O は、ストレージによって副サイトへコピーされる。リモートコピーには同期リモートコピーと非同期リモートコピーがあり、前者は正ストレージへの write I/O の延長で副ストレージへのコピーが実施されるのに対し、後者は正ストレージへの write I/O は正ストレージに書き込まれた時点で完了し、副ストレージへのコピーは別のタイミングで実施される。ストレージ主体の DR システムでは、ログ・DB とリモートコピーを用いて転送を行う。

リモートコピーを用いた DR システムでは、ストレージで転送が行われるため、アプリケーションや DBMS に転送ロジックを組み入れる必要がなく、システムが単純になるというメリットがある。また、転送にサーバリソースを使用しないので、サーバで転送を行う場合に比べてオンライン性能の確保も可能である。ただし、ログに加えて DB 本体の転送も必要となるため、転送量が大きくなる。一般に通信回線の維持コストは距離や帯域幅に応じ

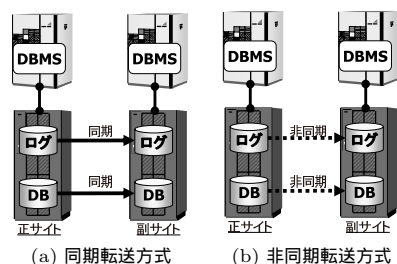


図 1 ストレージ主体の DR システム
Fig. 1 DR System using storage remote copy.

て高価になるため、更新量の大きな大規模 DB では多大な回線コストを平常時から負担しなければならない。

以下では、DBMS のログ、DB とも同期リモートコピー機能で転送する方式を同期転送方式 (図 1 (a))、ログ・DB とも非同期リモートコピーで転送する方式を非同期転送方式 (図 1 (b)) と定義する。同期転送方式では、正ストレージへの書き込みが完了した write I/O は副ストレージにもコピーされていることが保障されるため、災害時にも欠損なくデータを回復できるが、転送遅延が大きい場合は正サイト DBMS のオンライン性能も大きく低下する場合がある。一方、非同期転送方式では、副ストレージへの転送は正ストレージへの書き込みとは非同期に行われるため、正サイトのオンライン性能への影響は小さいが、災害が発生した場合にはデータ欠損が生じる可能性がある。

3.2 サーバ主体の DR システム

近年、多くのベンダが DBMS に転送機能を設けた製品を発表している^{18),19)}。これらの製品では、ログのみを副サイトに転送し、DB 本体は副サイトで正サイトから転送されたログを用いて更新する。ログベースの方式は、正サイトで発行したトランザクションは副サイトへの反映まで待って完了する 2-safe 方式、および、正サイトのトランザクションは副サイトでの適用とは非同期に完了する 1-safe 方式がある。なお、2-safe 方式には、ログだけを同期で転送しておき、更新 (反映) 自体は非同期で行う場合も含まれる。

サーバ主体の DR システムは、DB の定常的な転送が不要であり、転送量の削減が可能というメリットがある。しかし、転送処理の DBMS への組み込みが必要となるためシステムが複雑となる。また、転送処理にサーバリソースが必要となるため、正サイトのオンライン性能への影響もある。さらに、平常時から副サイトでログ適用を行うサーバを稼働させてお

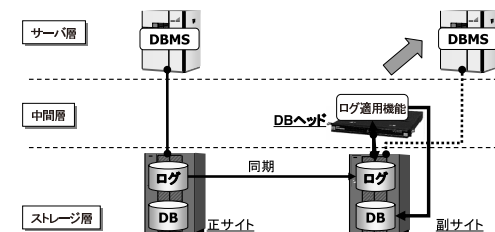


図 2 DB・ストレージ連携 DR システム (ログのみ同期転送方式)
Fig. 2 Log-based DR system using synchronous remote copy.

く必要があり、機器や運用コストが高くなるという問題もある。

3.3 DB・ストレージ連携による DR システム

ログの転送をストレージのリモートコピーで転送し、副 DB はログ適用で更新するシステムもある^{20)~22)}。たとえば、redo ログをリモートコピーで転送し、アーカイブしたログをサーバ間で転送・適用するシステムが知られている²⁰⁾。ただし、これらのシステムでも平常時の副 DB サーバの稼働は必要となる。

我々は、副サイトでのログ適用処理の負荷は正サイトのオンライン処理に比べると小さいことに着目し、ログ適用処理はリソースの限定された DR 向けアプライアンス (DB ヘッド) で実施する方法を提案してきた^{23),24)}。平常時は安価なアプライアンスを利用することで、平常時の機器・運用コストを削減することが可能となる。以下では、ログを同期リモートコピーで転送し、副サイトでのログ適用は DB ヘッドにより実現する方式をログのみ同期転送方式と呼ぶ (図 2)。

3.4 評価の方針

日本では地震など広域災害への対応が必要なこと、また、金融業界などでは災害時にもデータ欠損がないことが大きな価値となる。そこで、本論文では、同期転送を用いた方式を評価対象とした。また、正サイトのオンライン性能確保が実用上きわめて重要であるという観点から、ストレージの同期リモートコピーを用いた同期転送方式とログのみ同期転送方式を評価対象とした。

同期転送方式とログのみ同期転送方式を比較した場合、回復時間については、両者は同等である。DR では、平常時は副 DB サーバを起動させていないコールドスタンバイ方式をとるのが通常であり、災害時には、最初に副 DB サーバの起動を行う必要がある。そのため、災害時には、副 DB サーバの起動と、それに続いて副 DBMS の起動が必要となるが、これ

らの処理は同期転送方式、ログのみ同期転送方式とも必要となる。

回線コストに関しては、ログのみ同期転送方式が同期転送方式より優れる。これは、ログのみ同期転送方式は、平常時はログのみを転送しており、DB の転送が不要となるため、安価な狭帯域幅回線が利用可能となるためである。

一方、副サイトにおける構築、および、運用管理コストに関しては、同期転送方式が優れる。これは、同期転送方式は、平常時は副ストレージのみがあればよいのに対して、ログのみ同期転送方式では、副ストレージと DB ヘッドが必要となるためである。

以上のように両方式には、回線コストと副サイトのコストという観点では、それぞれ一長一短がある。ここで、既存のシステムに DR を適用するためには、サービスを提供している正サイトの運用が容易なことが求められる。特に DR 適用による性能面への影響を最小限に抑えることが必須である。そのため、ログの転送遅延やシステムの急激な負荷変動といった環境の変動に対しても安定した性能を維持することが求められる。そこで、本論文では、DR システムの性能をモデル化するとともに、回線シミュレータを用いた DR 環境で実験を行い、各システムの性能安定性を評価する。

4. 広域災害向けデータ欠損なし DR システムの性能安定性の考察

ログ出力と DB の更新処理に転送遅延の影響が及ぶ同期転送方式に比べて、ログのみ同期転送方式は、転送遅延の影響が及ぶ範囲がログ出力に限定されるためシステムがシンプルとなり、高い性能安定性を示すと考えられる。そこで、両方式の性能安定性を考察するために、転送遅延が DBMS 内部に及ぼす影響まで考慮した性能モデルを構築する。本章では、最初に、前提となる DB システムについて説明した後、構築した性能モデルについて述べる。そして、最後に性能モデルをもとに DR システムの性能安定性に対して考察を行う。

4.1 前提とする DB システム

前提とした DB システムの構成を図 3 に示す。トランザクション（以下、Tr）が行データを更新しようとするときには、まず WAL (Write-ahead log)²⁵⁾ に基づいてログの生成を行う。ここで生成されたログをログレコードと呼ぶ。生成されたログレコードはストレージに格納する必要があるが、更新のたびにストレージに書き出すのではなく、ログバッファにコピーしておき、Tr がコミットされたとき、あるいは、ログバッファが満杯になったときにストレージへフラッシュする。このログバッファは多面管理され、あるログバッファをストレージに書き出している最中、すなわち、ストレージからの完了通知 (Ack) を待っている間にも、他のログバッファを使うことで処理を継続できる。

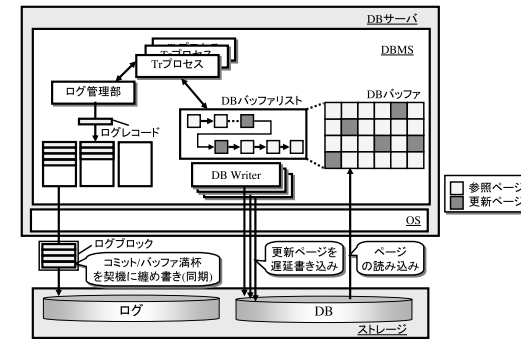


図 3 前提とする DB システム

Fig. 3 Target DB system for evaluation.

DB はページ単位で管理される。このページは DB ページと呼ばれ、内部に行データが格納される。Tr 実行中に行データへの参照/更新要求があった場合には、最初に、ストレージから該当する DB ページをサーバのメモリ上に確保された DB バッファ上に読み込んだうえで処理を行う。ここで、参照用に読み込んだ DB ページを参照ページ、更新用に読み込んだ DB ページを更新ページと呼ぶ。これらの DB ページは、DB バッファリストを用いることで、アクセス頻度に基づいた管理が行われる。

DB バッファへのアクセスがあると、アクセスされた DB ページはアクセス順に DB バッファリストに追加される。DB バッファリストに追加された DB ページが、ストレージに出力される前に再びアクセスされた場合は、該当ページをリストの先頭へ付け替える。DB バッファ上にないページをストレージから新たに読み込む場合には、空きページを取得する。DB バッファに空きページがない場合には、DB バッファリストの最後尾のページを空きページ (スロット) とした後に、新たなページを読み込む。このとき、最後尾の入れ替え対象の DB ページが参照ページであるならば、該当ページを破棄して空きページとした後に、新しいページを読み込む。一方、最後尾の DB ページが更新ページの場合は、まず、該当ページをストレージに出力して空きページとした後に、新しいページを読み込む。

更新ページの出力は DB 処理性能の低下の原因となる。そのため、DB Writer が、この DB バッファリストの更新ページ量を一定間隔で常時監視し、その値が一定の閾値を超えると、リスト中の更新ページをストレージに出力する。出力対象の更新ページは DB バッファリストの最後尾から順に選択される。ストレージに出力済みの更新ページは参照ペー

ジとして、DB バッファリストに残されるが、該当ページのリスト先頭への付け替えは行わない。このようにすることで、DB バッファリストの最後尾付近のページを参照ページに保ち、DB バッファ上に新たなページを読み込む際の更新ページのストレージへの出力動作を抑止する。なお、DB Writer は多重度をパラメータとして有しており、更新ページを多重出力することが可能であるとする。

なお、DBMS では一定時間経過後、あるいは、一定量の更新を行った後にチェックポイント処理²⁵⁾を行うが、チェックポイント処理は DBMS ごとに処理方式が異なるためモデル化の対象外とした。

OS は、AP や DBMS から複数の I/O 要求を受け付け可能であるとし、ストレージへの同時 I/O 発行数を表す QueueDepth をボリューム (VOL) ごとに設定可能とする。また、ストレージは、同期リモートコピー機能を有することを前提とする。

4.2 DR システムの性能モデル

DR システムの性能安定性を考察するために、転送遅延が DBMS 内部挙動に及ぼす影響を考慮して性能モデルを構築する。具体的には、DR を行わない local 構成の DBMS の挙動をベースとし、転送遅延がログ出力や DB 出力時に及ぼすペナルティを加えていくことで性能モデルを構築する。

4.2.1 ログのみ同期転送方式の性能モデル

ログのみ同期転送方式では、ログのみが転送遅延の影響を受ける。そのため、ストレージへの出力をとまなうコミットには必ず転送遅延 (L_{log}) だけのペナルティが加わる。さらに、ある Tr のコミット情報をストレージへ要求したときに、先行するコミット出力が完了していない場合は、その完了も待たなければならない。すなわち、ログバッファにログレコードを充填しておき、先行するログ出力が完了したときに、充填したログをまとめて出力する。これは、待合バスなどと同じ動作であり、待っている客を 1 度に処理 (運搬) するクリアリングシステムと見なせる。ここで、時刻 0 から時刻 T までに N 台のバスが到着した場合で、到着時刻を $t_1, t_2, \dots, t_n$ 、到着間隔を $x_1, x_2, \dots, x_n$ 、乗客は単位時間あたり a 人到着するとする。このとき、 k 台目のバスに乗る乗客は約 ax_k 人、それらの人の平均待ち時間は $x_k/2$ と見なせる。したがって、 N 台のバスに乗る aT 人の平均待ち時間は、以下の式で表される。

$$w = \frac{1}{aT} \sum_{k=1}^N (ax_k \times \frac{x_k}{2})$$

$$= \frac{1}{T} \sum_{k=1}^N \frac{x_k^2}{2} = \frac{N}{T} \frac{1}{N} \sum_{k=1}^N \frac{x_k^2}{2}$$

ここで、バスの到着間隔を確率変数 X で表すと、バスの平均到着間隔 $E(X) = m$ は、 T/N と見なせる。 $1/N \sum_{k=1}^N x_k^2$ は N が十分に大きいときに大数の法則で、 X^2 の期待値 $E(X^2) = V(X) + E(X)^2 = \sigma^2 + m^2$ と置き換えることができる。ここで、 $V(X) = \sigma^2$ は分散を表す。以上より、バスの平均待ち時間は以下の式に書き換えることができる。

$$w = \frac{1}{m} \frac{E(X^2)}{2} = \frac{1}{2m} (\sigma^2 + m^2) = \frac{m}{2} \left(1 + \left(\frac{\sigma}{m} \right)^2 \right)$$

ここで、 σ は回線品質によってばらつくことがある。しかし、専用回線など安定した回線ではばらつきは小さくなるため、本論文では $\sigma \simeq 0$ とした。その場合、待ち時間は転送遅延の平均 (L_{log}) に 0.5 を乗じた値となることから、コミット時の待ち時間は、転送時間 (L_{log}) に待ち時間 $L_{log} \times 0.5$ を加えた $L_{log} \times 1.5$ となる。

したがって、ログのみ同期利用時の Tr の応答時間 (R_{log}) は、DR を行わない場合の応答時間 (R_{local}) に、コミット時のペナルティ (P_{cmt}) を加えることによって求められる。 P_{cmt} は自コミットの出力時に生じる待ち L_{log} と先行コミット出力完了待ち $L_{log} \times 0.5$ を加えた $L_{log} \times 1.5$ となる。

$$\begin{aligned} R_{log} &= R_{local} + P_{cmt} \\ &= R_{local} + L_{log} \times 1.5 \end{aligned} \quad (1)$$

ここで、ユーザの思考時間とキー入力時間の和で表される Tr の投入間隔を $T_{interval}$ とすると、単位時間あたりに実行されるトランザクション数は、 $1/(T_{interval} + R_{log})$ となる。このようなトランザクションを実行する AP が N_{AP} 多重で実行されている場合、ログのみ同期のスループット T_{log} は、以下の式で表される。

$$T_{log} = N_{AP} / (T_{interval} + R_{log}) \quad (2)$$

ここで、多重度を増加させると、個々の Tr の応答時間は大きくなる。すなわち、スループットの式には、応答時間のファクタも含まれていることから、多重度を増大させると無限にスループットが向上するわけではなく、その上限値はシステムの最大スループットを超えない。

4.2.2 同期転送方式の性能モデル

同期転送方式は、ログ・DB とも同期リモートコピーで転送する。ログのみ同期転送方式との差異は、DB VOL の同期リモートコピーに起因し、転送遅延の影響で DB VOL の出

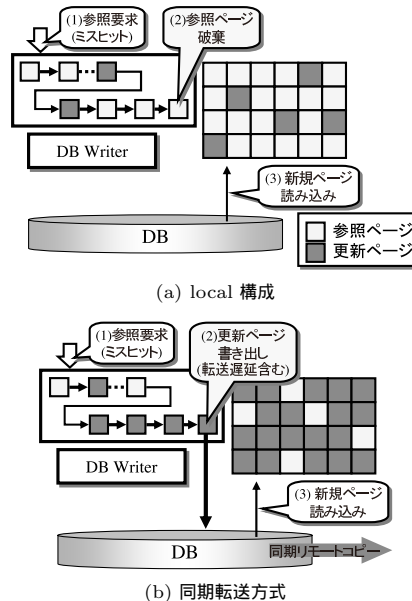


図 4 DR 環境での DB バッファの挙動
Fig. 4 Behavior of DB Buffer in DR system.

カスループットが低下した場合に性能が低下する点である。

同期転送方式では、転送遅延の影響で DB バッファ層での挙動が DR を行わない local 構成とは異なってくる。図 4 に local 構成と同期転送方式の DB バッファの挙動の概略図を示す。最初に、local 構成の参照ページの入力要求時の挙動について説明する。DB バッファ上の DB ページは、アクセス頻度を考慮するために、DB バッファリストで管理される。すなわち、ストレージから読み込んだ DB ページは参照ページとして DB バッファリストの先頭に追加される。ここで、Tr が更新処理を実行すると、DB バッファに読み込まれた参照ページ内の行データが更新され、更新ページとなる。ただし、この段階では更新ページのストレージへの書き込みは実行されない。更新ページのストレージへの書き込みは DB Writer が行う。DB Writer は、DB バッファリストを監視し、更新ページの量が一定量を超過した場合には、古くなった更新ページから順にストレージに出力する。そのため、DB Writer による更新ページの出力スループットが、Tr による更新ページ生成スループットより

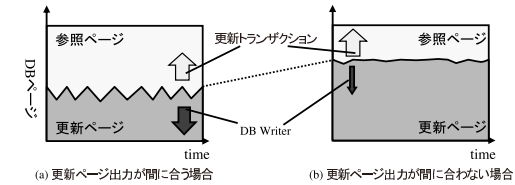


図 5 更新ページ比率の挙動
Fig. 5 Ratio of update page on DB buffer.

り大きければ、DB バッファ上では参照ページが多数を占めることになる（図 5(a)）。逆に、Tr による更新ページ生成スループットが、DB Writer による更新ページ出力スループットより大きければ、DB バッファ上では更新ページの割合が高まる（図 5(b)）。

DR を行わない local 構成では、DB Writer が正常に動作していれば図 5(a) に示すように DB バッファ上で更新ページの割合は低い値に保たれている。この状態で、DB ページのアクセス要求があって、かつ、この DB ページがバッファ上には存在しなかった場合は、まず空きページを確保してからストレージより該当ページを読み込む（図 4(a)）。具体的には、DB バッファリストの最後尾のページが入れ替え対象であり、このページが参照ページならば、破棄（ページ）を行うことで空きページを作成し、続いて、ストレージから要求されたページを読み込む。

図 4(b) に同期転送方式の実行時で、かつ、転送遅延の影響で更新ページの出力が間に合わない場合の DB バッファ挙動を示す。この場合、転送遅延の影響で更新ページの出力スループットが低下するため、DB バッファ上での更新ページの割合が高まる（図 5(b)）。そのため、DB バッファリスト最後尾においても更新ページの割合が高まる。

更新ページの割合が高まった状態（図 4(b)）で、DB ページのアクセス要求がなされ、かつ、この DB ページはバッファ上にはない場合には、空きページ確保に続いて該当 DB ページの読み込みが行われる。空きページ確保においては、DB バッファリスト最後尾の DB ページが入れ替え対象となるが、この DB ページがストレージへの反映が未済みな更新ページであった場合は、まず、この更新ページをストレージに出力しなければならない。以下、この処理を更新ページフラッシュと呼ぶ。同期転送方式では、DB VOL は同期リモートコピーでペアリングされているため、この出力処理には転送遅延分の応答時間が積み上げられてしまう。DB Writer による更新ページの出力は、Tr とは非同期で実施されるのに対し、DB ページ入力要求時に実施される更新ページフラッシュは Tr の一処理として実行されるため、

Tr 性能に大きく影響する．

DB Writer の多重度を上げて更新ページの出力スループットを向上させることで、この影響を小さくすることができる．ここで、ログは WAL²⁵⁾ に基づいて出力されるため、更新ページを出力する場合には、関連するログが事前にストレージに出力されていなければならないが、通常は更新ページの出力がログ出力に待たされることはない．これは、性能向上の観点から、DB Writer は、ログ出力から遅れて該当する更新ページの出力を行うように実装されるためである．すなわち、性能向上実現のためには、なるべく DB バッファ上で複数の更新をまとめ、ストレージへの出力回数を最小限にすることが望ましい．そのため、DB Writer では、たとえば、更新ページの数が一定量（閾値）を超過したときにのみ、超過分を出力する．このような実装の下では、ログ出力をとまなうコミット処理から遅れて関連する更新ページが出力されるのが通常となるので、更新ページの出力がログ出力の完了を待つことはない．

しかし、DB Writer の多重度を大きくしすぎた場合にも性能は低下する．OS は I/O 要求をリクエストキューに格納し、このキューから QueueDepth 分の要求を取り出してストレージに対して多重発行する．そのため、更新ページの出力スループット向上のために DB Writer の多重度を大きくしすぎた場合には、リクエストキュー内で I/O が滞留して待ちが生じる．このような滞留が生じると read 処理にも待ち時間だけペナルティが加わる．

以上をまとめると、同期転送方式では、転送遅延の影響で DB 出力が間に合わない場合に性能が低下する．具体的には、DB Writer 多重度が不足する場合であり、このとき、空きページ確保のために必要となる、更新ページの出力（更新ページフラッシュ）によって性能が低下する．逆に DB Writer 多重度を過度に大きくしすぎた場合にも、OS 内で I/O が滞留し、その待ち時間の影響で性能が低下する．そのため、DB Writer 多重度には最適値が存在し、適度な値を設定した場合には、DB バッファの挙動は local 構成やログのみ同期と同等となる．ただし、ストレージ上のログ VOL についても同期リモートコピーでベアリングを行っているため、同期転送方式の性能はログのみ同期転送方式の性能を超えることはない．したがって、同期転送方式のスループット性能は、(1) ログのみ同期方式のモデル式、(2) DB Writer 多重度不足時のモデル式、(3) DB Writer 多重度過多時のモデル式の最小値で表される（図 6）．図 6 より同期転送方式の性能は DB Writer 多重度に対して凸型のグラフとなるため、最適性能を出すためには、local 構成時には不要だった DR 向けのチューニングが必要となる．

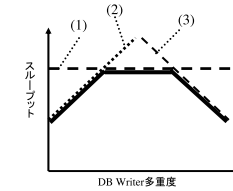


図 6 同期転送方式の性能モデル

Fig.6 Performance model of DR system (Sync Transfer).

(1) ログのみ同期転送方式の性能モデル

図中の (1) はログのみ同期転送方式のモデル式（スループット）で表される（式 (2)）．

(2) 同期転送方式の性能モデル 1

図中 (2) は同期転送方式の DB Writer 多重度不足時の性能を表しており、このときのトランザクション応答時間 R_{sync} は、local 時の応答時間 R_{local} に、コミット時のペナルティ P_{cmt} と更新ページフラッシュのペナルティ P_{wflush} を加えた値となる． P_{wflush} は、バッファミスをとまなうページの入力があり、かつ、DB バッファリストの最後尾が更新ページであった場合に生じる．このときの確率は、バッファヒット率を P_{hit} とし、DB バッファリスト内の最後尾のダーティ比率を P_{upd} とした場合、 $(1 - P_{hit}) \times P_{upd}$ と表せる．したがって、Tr あたりの（参照・更新）ページ入力要求回数を N_{get} とし、DB VOL に対する write 応答時間を L_{db} とした場合、更新ページフラッシュのペナルティは以下の式で表される．

$$P_{wflush} = (1 - P_{hit}) \times P_{upd} \times L_{db} \times N_{get}$$

ここで、local 構成時に単位時間あたりにストレージに出力される更新ページ数を N_{wpage} 、単位時間あたりに DB バッファにミスヒットするページ数を $N_{miss} (= (1 - P_{hit}) \times N_{get} \times T_{local})$ とする．すると同期転送方式のときの単位時間あたりの更新ページ出力数 N'_{wpage} と DB バッファへのミスヒット率 N'_{miss} はそれぞれ以下の式で表される．

$$N'_{wpage} = N_{wpage} \times (T_{sync}/T_{local})$$

$$N'_{miss} = N_{miss} \times (T_{sync}/T_{local})$$

ここで、 T_{local} は、DR を行わない local 構成時の Tr スループットであり、 T_{sync} は同期転送方式を行うときの Tr スループットを表す．もし、DB Writer 多重度が 0 であるならば、すなわち、DB Writer を使わなければ、 N'_{wpage} 個のページはすべて更新ページフラッシュによって出力しなければならない．このときの DB バッファリスト最後尾のダーティ率 P_{upd} は $P_{upd} = N'_{wpage}/N'_{miss} = N_{wpage}/N_{miss}$ となる．

続いて, DB Writer の出力が間に合うようになり, 更新ページフラッシュが発生しなくなるときの P_{upd} を計算する. ここで, 単位時間あたり N'_{wpage} 個の更新ページの出力が可能となるとき DB Writer 多重度を N_{prc} とするならば, $N'_{wpage} \leq N_{prc}/L_{db}$ が成り立つ.

このとき, N'_{wpage} は $N_{wpage} \times (T_{sync}/T_{local})$ と表せたが, DB Writer の出力が間に合っており DB 出力によるペナルティがなくなっている状態では, $T_{sync} = T_{local}$ と見なせるため, N'_{page} は N_{page} に以下に示す係数 k を乗じることで計算できる.

$$k = \frac{T_{sync}}{T_{local}} = \frac{T_{log}}{T_{local}} = \frac{T_{interval} + R_{local}}{T_{interval} + R_{local} + 1.5 \times L_{log}}$$

そのため, $0 < N_{prc} < N_{wpage} \times L_{db} \times k$ の間で線形補完すると P_{upd} は次のような N_{prc} の関数で表すことができる.

$$P_{upd} = \begin{cases} (1 - \frac{N_{prc}}{L_{db} \times N_{wpage} \times k}) \times \frac{N_{wpage}}{N_{miss}} & 0 \leq N_{prc} \leq N_{wpage} \times L_{db} \times k \\ \dots 0 & 0 \leq N_{prc} \leq N_{wpage} \times L_{db} \times k \\ 0 \dots N_{prc} > N_{wpage} \times L_{db} \times k \end{cases}$$

したがって, DB Writer 多重度不足時の同期転送方式の応答時間 R_{sync} とスループット T_{sync} は次の式で表される.

$$\begin{aligned} R_{sync} &= R_{local} + P_{cmt} + P_{wflush} \\ &= R_{local} + L_{log} \times 1.5 \\ &\quad + (1 - P_{hit}) \times P_{upd} \times L_{db} \times N_{get} \\ &= R_{local} + L_{log} \times 1.5 + (1 - P_{hit}) \\ &\quad \times (1 - \frac{N_{prc}}{(L_{db} \times N_{wpage} \times k)}) \\ &\quad \times \frac{N_{wpage}}{N_{miss}} \times L_{db} \times N_{get} \end{aligned} \quad (3)$$

$$T_{sync} = N_{AP} / (T_{interval} + R_{sync}) \quad (4)$$

(3) 同期転送方式の性能モデル 2

DB Writer 多重度を適切な値以上に設定した場合は, 更新ページフラッシュは生じなくなる. しかし, 多重度を大きくしすぎると, 転送遅延以上の応答時間を要する複数の write I/O の発行・完了待ちをしなければならず, OS 内で待ちが生じる. ここで, DB Writer 多重度過多時には更新ページフラッシュは発生せず, また, 更新ページの出力は Tr とは非同

期で実施されるため, 更新ページの出力処理は Tr 応答時間に影響しない. そのため, OS 層での待ちは read 処理, すなわち, 参照ページの要求があり, かつ, バッファミスしたときのみ Tr 応答時間へ影響する. したがって, OS 層での待ちによって生じるペナルティ P_{os} は, 参照ページ要求がヒットする率を P_{refhit} , Tr あたりの参照ページ要求回数を N_{refget} , OS 層での待ち W_{os} を用いて, 以下の式で計算される.

$$P_{os} = (1 - P_{refhit}) \times W_{os} \times N_{refget}$$

また, W_{os} については, OS の QueueDepth (q_{depth}) を超える多重 I/O が発行された場合に待ちが生じることから, 以下の式で計算することができる. なお, A は環境ごとに定まる定数であり, たとえば, ランダム I/O を行うテストプログラムを多重実行することで求められる.

$$W_{os} = \begin{cases} 0 & (N_{prc} \leq q_{depth}) \\ A(N_{prc} - q_{depth}) & (N_{prc} > q_{depth}) \end{cases}$$

したがって, DB Writer 多重度過多時の同期転送方式の応答時間とスループットは以下の式で表される.

$$\begin{aligned} R_{sync} &= R_{local} + P_{cmt} + P_{os} \\ &= R_{local} + L_{log} \times 1.5 \\ &\quad + (1 - P_{refhit}) \times W_{os} \times N_{refget} \end{aligned} \quad (5)$$

$$T_{sync} = N_{AP} / (T_{interval} + R_{sync}) \quad (6)$$

4.2.3 DR システムの安定性の考察

DB VOL に転送遅延の影響が及ぶ同期転送方式は, DB Writer の多重度が重要なパラメータとなる. そのため, DB Writer の多重度を横軸にした場合, オンライン性能は凸型のグラフになる (図 7(a)).

最適な DB Writer 多重度は, 回線遅延といった環境要因にも影響を受けるため, 環境変

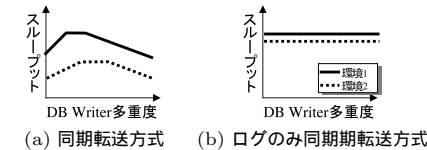


図 7 DR システムの性能安定性の考察
Fig. 7 Study on performance stability of DR system.

動があるたびにチューニングが必要となり、性能安定性は低くなる。一方、DB の出力には転送遅延の影響を受けないログのみ同期転送方式は、システムを単純化することができ、環境変動に対しても、local 構成でのチューニングが流用でき、高い安定性を示す（図 7(b)）。

5. DR システムの性能安定性の評価

本章では、回線シミュレータを用いた DR 環境で実験を行い、DBMS 内部挙動が 4 章での仮説どおりの挙動を示していることを確認するとともに性能モデルの予測精度を確かめる。さらに、性能安定性を評価するために、回線遅延や回線帯域幅などの環境因子を変動させたときの各 DR 方式の安定性を評価する。

5.1 評価環境と入力負荷モデル

実験に用いた評価環境を図 8 に示す。評価環境では、正 DB サーバと正ストレージを正サイトと見立て、同様に副ストレージを副サイトと見立て、サイト間は正副ストレージをエクステンダと回線シミュレータを介して接続した。評価に用いた実験機器、OS、ミドルウェアを表 1 に示す。

実験では AP サーバから正 DB サーバ上の DBMS に負荷を投入し、正 DBMS のオンライン性能として単位時間あたりの処理トランザクション数や DBMS 内部の統計情報を計測した。なお、定常状態での評価を行うために、測定は 1 分のウォームアップを行った後に行った。

負荷モデルは、TPC-C²⁶⁾ ベースの販売管理モデルを利用し、WH 数は 32 としてデータ生成を行った。なお、実験を単純化するために Neworder トランザクションのみを用いて負荷を生成した。具体的には、Neworder トランザクションを発行するプログラムを用意し、これらを AP サーバ上で多重実行した。なお、入力負荷の多重度や投入間隔は、DBMS 内の論理的な排他がネックとならない範囲で設定した。負荷モデル、DBMS、OS のパラメータを表 2 に示す。以降で実施した実験では、特にことわりがない場合には、表 2 のパラメータを用いた。

5.2 性能モデルの検証

性能モデルの有効性確認のために、DR 評価環境で実験を行い DBMS が 4 章で立てた仮説どおりの挙動を示しているかどうかを確認した。

5.2.1 ログ出力の挙動

ログ出力の挙動を明らかにするために、転送遅延をパラメータとしてログのみ同期転送方式と同期転送方式を評価した。なお、DB Writer 多重度は 1 として実験を行った。最初に、

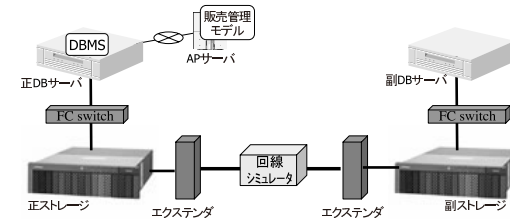


図 8 DR 評価環境

Fig. 8 Evaluation environment.

表 1 DR 評価環境で用いた実験機器

Table 1 Hardware which is used in evaluation.

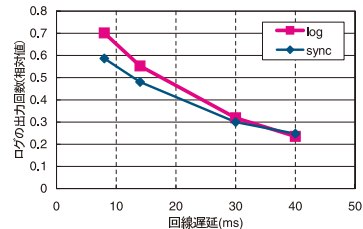
	製品
DB サーバ	Xeon2.40Hzx2, 1 GB, OS: linux-2.4.20, DBMS: HiRDB v7.2
ストレージ	SANRISE9570V (1 GB)
AP サーバ	Dell (Xeon2GHzx2) Win2K AS
エクステンダ	McData Eclipse1620
回線シミュレータ	Empirix PacketSphere (100 Mbps)
FC switch	Brocade 3200

表 2 負荷モデル・DBMS・OS の主なパラメータ

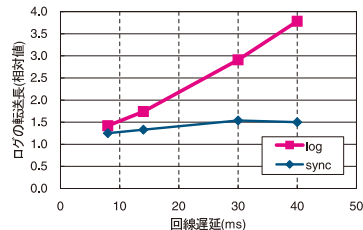
Table 2 Parameter of Workload/DBMS/OS.

パラメータ		設定値
負荷モデル	WH 数	32
	投入間隔	0.63
	多重度	32
DBMS	ページサイズ	4 KB
	DB バッファ面数	6000
OS	QueueDepth	16

ログの出力回数と出力あたりのブロックサイズ（出力長）と転送遅延の間の関係性を評価した。結果を図 9 に示す。ここで、ログの出力回数、出力長とも DR を行わない local 構成時の値に対する相対値で示した。これらの値は、正サイトのストレージへの出力に関する統計情報であるため、DR の有無にかかわらず、DBMS が稼動するサーバにおいて計測できる。図 9 より、ログのみ同期転送方式（log）では、転送遅延が大きくなるに従って単位時



(a) ログの出力回数



(b) ログの出力長

図 9 ログの出力回数と出力長

Fig. 9 Frequency and size of log writing.

間あたりの出力回数は減少するが、逆に出力長は大きくなっており、転送遅延が大きくなった場合でも、ログ出力回数と出力長の積から計算されるログ出力スループットの確保は可能であることが分かる。ここで、Tr スループットとログ出力スループットは比例するため、Tr スループットが確保されたといえる。これは、DBMS 内部でログバッファが多面管理されており、あるログバッファの出力待ちの間も他のバッファを用いて処理が継続されていくためである。

一方、同期転送方式 (sync) は、転送遅延を大きくするとログ出力回数は減少していくが、ログの出力長は一定であるため、転送遅延を大きくするとスループットは低下する。これは、同期転送方式では、DB VOL も同期リモートコピーでペアリングしているため、DB VOL の出力性能が転送遅延の影響で低下し、それが DB バッファ挙動にも影響していくことで Tr 処理も遅延していくためである。以降の節では、同期転送方式実行時の DB バッファ挙動について説明する。最初に、DB Writer 多重度が不足した場合について説明した後、DB Writer 多重度が過剰になった場合について説明する。

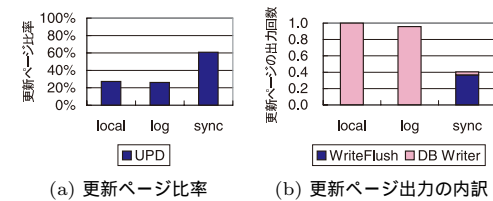


図 10 DB バッファの挙動

Fig. 10 Behavior of DB buffer.

5.2.2 DB バッファの挙動 1 (多重度不足時)

同期転送方式の内部挙動の特徴を明確化するために、各システム (local/ログのみ同期転送方式/同期転送方式) の DB バッファ層の挙動を比較した。具体的には、片道の転送遅延を 15 ms、DB Writer 多重度を 1 として各システムの実験を行い、DB バッファ内の更新ページの比率、および、更新ページ出力処理の内訳を調査した。結果を図 10 に示す。なお、図 10(b) については、各 DR システムの DB ページの出力回数は、local 構成の DB ページ出力回数に対する相対値とした。

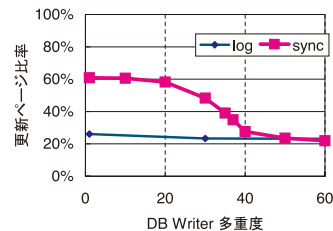
図 10(a) より、local とログのみ同期転送方式 (log) については、更新ページ (UPD) の比率は約 3 割で同等となっているのに対し、同期転送方式 (sync) では更新ページ比率は local/ログのみ同期転送方式時の 2 倍以上になっており、DB Writer による更新ページの出力が間に合っていないことが分かる。

図 10(b) には、更新ページ出力処理の内訳を比較した。DB Writer の出力が間に合っている場合は、更新ページは DB Writer によって出力される。一方、転送遅延の影響で DB Writer の出力が間に合わなくなった場合は、前述のように、DB バッファリスト内の更新ページ比率が高まっていく結果、空きページ確保のためにリスト最後尾の更新ページを出力する更新ページフラッシュが生じる。すなわち、更新ページは、DB Writer が更新ページフラッシュのいずれかによって出力されるが、更新ページフラッシュは Tr 処理の延長で行われるため、Tr の性能が低下する。図 10(b) より、local とログのみ同期転送方式 (log) では、更新ページフラッシュ (WriteFlush) が発生しておらず、すべての更新ページは DB Writer によって出力されていることが分かる。一方、同期転送方式 (sync) では、更新ページ出力処理の 9 割以上が更新ページフラッシュによって実施されている。更新ページフラッシュによる Tr の応答時間の遅延が DBMS の全体処理性能の低下にもつながるため、同期転送方式の DB ページの出力量は local/ログのみ同期転送方式の 4 割に低下している。

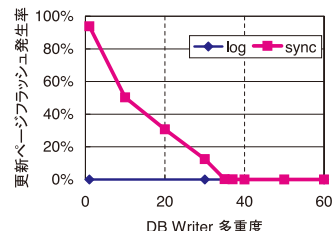
以上をまとめると、local とログのみ同期転送方式では DB バッファ層は同様の挙動を示しており、チューニングに関しても local 構成での方式が利用できる。一方、同期転送方式は、転送遅延の影響で DB Writer の出力が間に合わず、更新ページ比率が高まっていること、その結果、更新ページフラッシュが生じていることが確かめられた。したがって、同期転送方式は、DB バッファ層の挙動が他方式と異なっているため、チューニング方式も DR 独自の方式が必要となる。

同期転送方式は、DB Writer の多重度不足のときに転送遅延の影響で更新ページの出力が間に合わなくなり、その結果、更新ページフラッシュが生じて性能が低下する。しかし、DB Writer の多重度を適切な値にすれば、更新ページの出力が間に合うようになり、性能が向上すると考えられる。そのため、回線シミュレータの転送遅延を 15 ms (片道) に設定したうえで、DB Writer 多重度をパラメータとした実験を行い、同期転送方式における DB バッファ層の挙動を調査した。具体的には、DB バッファ内の更新ページ比率と DB Writer 多重度の関係、および、更新ページフラッシュ発生率と DB Writer 多重度の関係を調査した。

結果を図 11 に示す。図 11(a) より、DB Writer 多重度を増加させるにつれ、更新ペー



(a) 更新ページ比率



(b) 更新ページフラッシュ発生率

図 11 更新ページ比率と更新ページフラッシュ発生率

Fig. 11 Rate of update page and rate of update page flush.

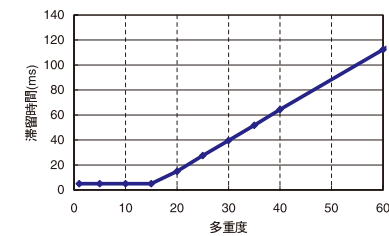
ジの比率は減っていき、最終的には同期転送方式 (sync) の更新ページ比率は local やログのみ同期転送方式 (log) と同等になることが分かる。また、図 11(b) より、DB Writer 多重度を大きくすることで、更新ページフラッシュの発生を抑えられることが確かめられた。

以上の結果、同期転送方式では転送遅延の影響で DB 出力が間に合わない場合に性能が低下するが、DB Writer 多重度のチューニングを行えば DB バッファ層の挙動も local と同等になり、性能も向上することが確かめられた。

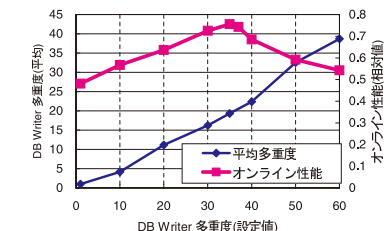
5.2.3 DB バッファの挙動 2 (多重度過多時)

多重度を大きくしすぎ、OS 層が受け付け可能な I/O 数を超過した場合には、OS 層で生じる待ちによって同期転送方式の性能は低下する。これを確かめるために、最初に、同期リモートコピーでペアリングされた VOL に対してランダム I/O を行うテストプログラムを用いて多重度と OS 内待ち時間 (滞留時間) との関係を調査した。その結果、QueueDepth より大きな値を多重度として設定した場合に、滞留時間が線形に増加していくことが確かめられた (図 12(a))。

DB Writer 多重度とオンライン性能の関係を図 12(b) に示す。図 12(b) より、同期転送



(a) Write I/O の滞留時間



(b) DB Writer 多重度とオンライン性能の関係

図 12 OS 層の挙動

Fig. 12 Behavior in OS layer.

方式の DB Writer 多重度を 35 とするまではオンライン性能は向上していくが、多重度を 35 以上にした場合は、性能が低下することが確かめられた。ここで、性能低下が始まる多重度は 35 であり、OS の QueueDepth (= 16) より大きくなっているが、これは、DB Writer の動作に起因する。すなわち、今回の実験で使った DB Writer は更新ページの数が増え、閾値を超過したときに起動され、閾値超過分を多重出力しようとする。そのため、起動直後は設定した多重度で動作するが、一定時間が経過し、出力すべき更新ページの数が増え、より少ない多重度での動作で間に合うためである。実際に、OS の Top コマンドを用いて平均的に起動されている DB Writer プロセスの数 (図 12 (b) 中の平均多重度) を監視してみると、性能低下が始まる平均多重度は 20 であり、OS の QueueDepth に近い値が得られている。

以上の実験から、性能モデル構築にあたって想定した内部挙動を実験によって確かめることができた。すなわち、ログのみ同期転送方式では、先行するログバッファの転送待ちの間も残りのログバッファを用いて処理を継続できるため、転送遅延が大きい場合にも、転送あたりの転送サイズ (ログバッファに溜めるログ量) を大きくすることで性能確保が可能となる。同期転送方式については、以下の 2 つのケースで性能が低下する。1 つ目のケースは、DB Writer 多重度が不足する場合であり、更新ページの出力が間に合わないため、Tr プロセスが要求した DB ページがバッファになかった場合に、更新ページフラッシュが発生する。この更新ページフラッシュは Tr の延長で行われるためオンライン性能が低下する。2 つ目のケースは、DB Writer 多重度過多の場合であり、OS で処理可能な I/O 量を超えた要求がなされると、OS 内での待ちが生じるため参照処理に遅延が生じ、オンライン性能が低下する。

5.2.4 性能モデルの検証

これまでの実験結果から、ログのみ同期転送方式と同期転送方式は、4 章での性能モデル構築にあたって想定した内部挙動を示すことが確かめられた。本項では、さらに、性能モデルから計算した予測値と実験実測値との比較を行うことによって、4 章で構築した性能モデルの有効性を検証した。最初に、ログのみ同期転送方式の性能モデルの検証を行い、続いて、同期転送方式の性能モデルの検証を行った。ここで、同期転送方式では DB Writer 多重度がキーとなるパラメータであることが分かっているため、性能モデルの検証においては、DB Writer 多重度をパラメータとして実験を実施し、性能予測値との比較を行った。

ログのみ同期転送方式の性能モデルは、local 構成での性能をベースとして、転送遅延のペナルティを加えることで計算される。そのため、最初に、local 構成での実験を行って必

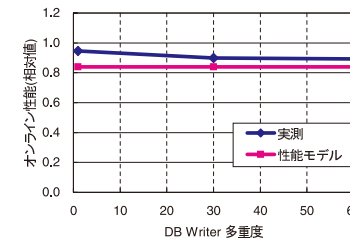


図 13 性能モデルの検証 1 (ログのみ同期転送方式)

Fig. 13 Verification of performance model (log).

要な統計情報を取得しておく必要がある。ここで、ログのみ同期転送方式の性能モデルは式 (1)、式 (2) で表されるため、local 構成での応答時間 (R_{local}) のみが必要となる。この値と転送遅延のみを用いて予測値を計算した。

続いて、ログのみ同期転送方式の環境 (回線シミュレータの転送遅延は 15 ms (片道)) を構築し、DB Writer 多重度をパラメータとして実験を行い、性能モデルから計算した予測値との比較を行った。結果を図 13 に示す。ここで、グラフ縦軸のオンライン性能は、local 構成時のオンライン性能に対する相対値とした。この結果、ログのみ同期転送方式は DB Writer 多重度によらず一定の性能が得られており性能安定性が高いことが実測値から確かめられた。また、実測と予測値との比較から、検討した性能モデルは約 1 割の誤差で性能予測が可能であることが確かめられた。

同期転送方式についても、式 (3)、(4)、(5)、(6) で示すように、local 環境での性能をベースに、更新ページフラッシュによって生じるペナルティや OS 層での滞留によって生じるペナルティを加算していくことで予測値を計算する。したがって、最初に、local 環境での実験結果から表 3 に示す統計値を取得しておく。これらの値と、ログ VOL の転送遅延、DB VOL の転送遅延、および、DB Writer 多重度の設定値を用いて、DB Writer 多重度ごとに予測値を計算する。

図 14 に同期転送方式を DB Writer 多重度をパラメータとして実験した実測値 (回線シミュレータ転送遅延は 15 ms (片道)) と、性能モデルから計算した予測値の比較結果を示す。なお、予測値に関しては、DB Writer 多重度不足のケースを表す式 (4) による予測値と DB Writer 多重度過多のケースをモデル化した式 (6) による予測値をそれぞれ示した。この 2 つの予測値の最小値が同期転送方式の予測値となる。この結果、同期転送方式は DB Writer 多重度に対し凸型のグラフとなり、最適性能を得るためにはチューニングが必要で

表 3 性能モデルのパラメータ値 (local 環境実測値)

Table 3 Parameter of performance model.

R_{local}	Tr (トランザクション) の応答時間
T_{local}	Tr のスループット
P_{hit}	バッファヒット率
P_{refhit}	参照ページ要求のヒット率
N_{get}	Tr あたりに含まれる参照・更新ページ要求回数
N_{refget}	Tr に含まれる参照ページ要求回数
N_{wpage}	単位時間あたりに出力される更新ページ数

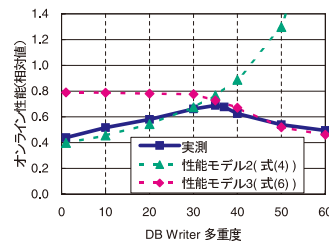


図 14 性能モデルの検証 2 (同期転送方式)

Fig. 14 Verification of performance model (sync).

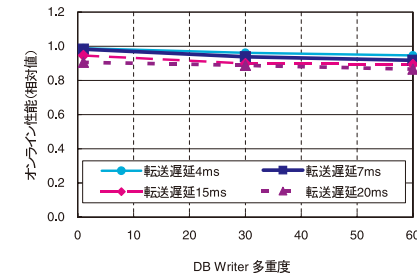
あり、ログのみ同期転送方式に比べて性能安定性が低いことが確かめられた。また、実測とモデルからの予測値との比較においては、検討した性能モデルは約 1 割の誤差で性能予測が可能ながことが確かめられた。

5.3 性能安定性の評価

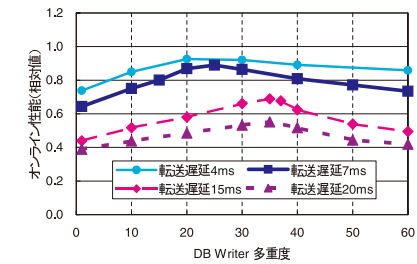
前節での検証から、ログのみ同期のモデル式は転送遅延の 1 次式であり DB Writer 多重度に対して頑強と考えられるのに対し、同期転送方式のモデル式は 3 つの式の最小値計算によって求められ凸型になることから、環境変動が生じると以前に実施したチューニングは有効でなくなり、最適性能が得られなくなると考えられる。そのため、転送遅延など環境パラメータを変動させながら実験を行い、ログのみ同期転送方式と同期転送方式の性能安定性を評価した。以下では、転送遅延/転送帯域幅/AP 入力負荷といった 3 つの環境パラメータを変動させた場合の性能安定性を評価した。

5.3.1 転送遅延が変動するケース

最初に転送遅延を変動させた場合について、ログのみ同期転送方式と同期転送方式の性能安定性を評価した。具体的には、転送遅延が異なる 4 つの環境 (4 ms, 7 ms, 15 ms, 20 ms)



(a) ログのみ同期転送方式



(b) 同期転送方式

図 15 性能安定性の評価 1 (転送遅延が変動するケース)

Fig. 15 Evaluation of performance stability1 (change of delay).

で、DB Writer 多重度をパラメータとしてログのみ同期転送方式と同期転送方式の正 DBMS オンライン性能を計測した。結果を図 15 に示す。

図 15(a) より、ログのみ同期転送方式では、異なる転送遅延であっても同等の性能が得られていることが分かる。これは、ログのみ同期転送方式では、転送遅延が大きい場合は、1 回あたりの転送サイズを大きくすることでスループットの確保が可能なたためである。また、DB Writer 多重度によらず、一定の値が得られており、DB Writer 多重度に対して性能安定性が高いといえる。一方、同期転送方式では、図 15(b) に示すように、転送遅延が変動するとオンライン性能も大きく変動しており、DB Writer 多重度が 1 のときで、4 ms から 20 ms に変動した場合では性能は約 1/2 に低下している。また、転送遅延が変動すると、ピーク性能が得られる DB Writer 多重度も変動してしまっている。たとえば、転送遅延が 7 ms から 15 ms に変動した場合では、ピーク性能を得るためには DB Writer 多重度を 10 だけ追加する必要がある。これは、転送遅延の大きい環境では、同一量の更新ページ

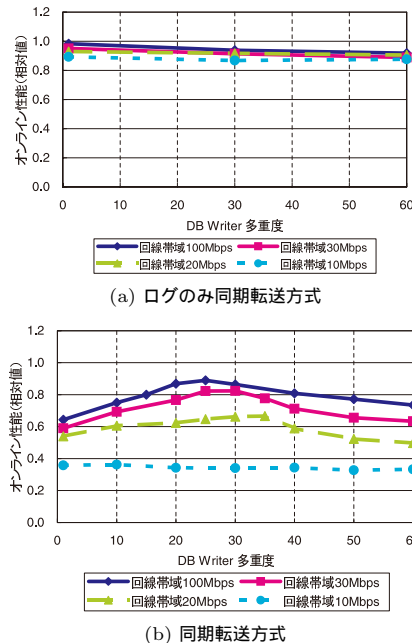


図 16 性能安定性の評価 2 (転送帯域幅が変動するケース)
Fig. 16 Evaluation of performance stability2 (band width).

を出力するのに、より多くの多重度が必要となるためである。したがって、同期転送方式では、公衆回線など品質保証のない回線を使っており、この回線を共有している他の AP の影響で転送遅延が変動してしまう場合には、以前に実施したチューニングパラメータは有効でなくなり、変動後の環境で最適性能を得るには新たなチューニングが必要になってしまう。

5.3.2 転送帯域幅が変動するケース

次に、回線帯域幅を変動させた場合について、ログのみ同期転送方式と同期転送方式の性能安定性を評価した。具体的には、回線帯域幅が異なる 4 つの環境 (100 M, 30 M, 20 M, 10 Mbps) で、DB Writer 多重度をパラメータとしてログのみ同期転送方式と同期転送方式の正 DBMS のオンライン性能を計測した。なお、回線遅延については、7 ms (片道) に固定したうえで実験を行った。結果を図 16 に示す。

図 16(a) より、ログのみ同期転送方式では、異なる転送帯域幅であっても同等の性能が

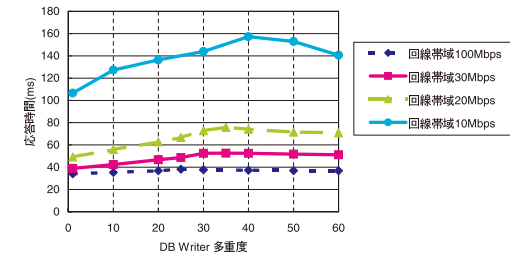


図 17 回線帯域幅と応答時間の関係
Fig. 17 Relation between band width and response time.

得られていることが分かる。また、DB Writer 多重度によらず、一定の値が得られており、DB Writer 多重度に対して性能安定性が高いといえる。一方、同期転送方式では、転送帯域幅が変動するとオンライン性能も変動している (図 16(b))。転送帯域幅が変動するとピーク性能が得られる DB Writer 多重度も変動してしまっており、転送帯域幅が 100 Mbps から 20 Mbps に変動した場合では、ピーク性能を得るためには DB Writer 多重度を 10 だけ追加する必要がある。

これは、帯域幅を狭めた回線に、そのキャパシティを超えるデータを転送しようとした場合、転送が待たされる場合があるため、平均的な応答時間は転送遅延を超過するためである。実際に、同期転送方式実施時に OS で write I/O の平均応答時間をグラフ化してみると、転送帯域幅を狭めるほど平均応答時間は悪化している (図 17)。そのため、帯域幅を狭めた場合ほど、OS や DBMS から見た応答時間は大きくなりオンライン性能は低下する。また、転送遅延が大きくなると同量データの転送には、より多くの多重度が必要となることから、回線帯域幅が狭まった環境においてピーク性能を得るためには、より大きな多重度が必要となる。具体的には、20 Mbps でチューニング実施後に回線の状況が回復して 30 Mbps, 100 Mbps になった環境で、以前の DB Writer 多重度のままで運用を行った場合には、本来得られるはずのピーク性能に対して約 7% の性能低下が生じている。したがって、公衆回線など品質保証のない回線を使っており、転送帯域幅が変動してしまう場合にも、同期転送方式では以前に実施したチューニングパラメータは有効でなくなり、変動後の環境で最適性能を得るには新たなチューニングが必要になってしまう。

5.3.3 入力負荷が変動するケース

続いて、AP の入力負荷を変動させた場合について、ログのみ同期転送方式と同期転送方式の性能安定性を評価した。具体的には、図 18 に示すようなスケジュールで負荷を投入す

ることで、負荷集中期間を作り出し、負荷集中時の性能安定性を評価した。すなわち、実験期間を通じてベースとなる負荷をかけておき、中間期間には追加負荷を投入することで負荷集中期間を生成した。なお、実験を単純化するために入力負荷にも変更を加えた。Newoder トランザクションは、District、Stock 表への更新と Orderline、Orders、NewOrder 表への挿入を含んでいるが、変更した負荷では Stock 表への更新のみを行うようにした。また、負荷投入スケジュールについては、ベース負荷を 16 多重とし、負荷集中期間には合計負荷

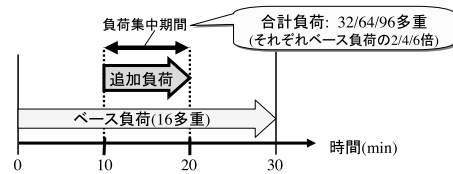


図 18 負荷投入のスケジュール
Fig. 18 Schedule of workload.

がベース負荷の 2/4/6 倍 (32/64/96 多重) になるように負荷をかけた。以下、それぞれの負荷を負荷 1/負荷 2/負荷 3 と呼ぶ。図 19、図 20、図 21 に、各 DR システムにこれらのスケジュールで負荷を投入したときのトランザクション応答時間を示す。なお、同期転送方式に関しては、合計多重度が 32 (ベース負荷の 2 倍) の環境でチューニングを行い、各実験を通じて DB Writer 多重度はこのときのチューニング値 (10) で固定とした。

図 19 より、集中時にベースの 2 倍の負荷がかかる場合 (負荷 1) では、各 DR システムの性能は local と同様の挙動を示し、負荷集中期間であっても安定した性能が得られていることが分かる。一方、図 20 では、集中時にはベースの 4 倍の負荷をかけているが (負荷 2)、このときは、ログのみ同期転送方式では負荷集中時にもほとんど応答時間は悪化していないのに対し、同期転送方式は、負荷集中時にはばらつきが大きくなり、応答時間が 2.0 sec を超える場合も見られる。さらに、負荷集中時にベースの 6 倍の負荷をかけた場合 (負荷 3、図 21) には、同期転送方式では負荷集中時のばらつきはさらに増大しており、応答時間が

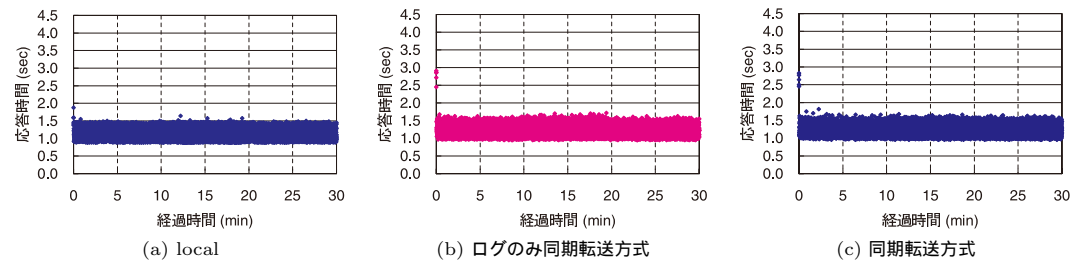


図 19 性能安定性の評価 3 (入力負荷が変動するケース、集中時: ベース負荷 $\times$ 2)

Fig. 19 Evaluation of performance stability3 (fluctuation of work load, peak load: base load $\times$ 2).

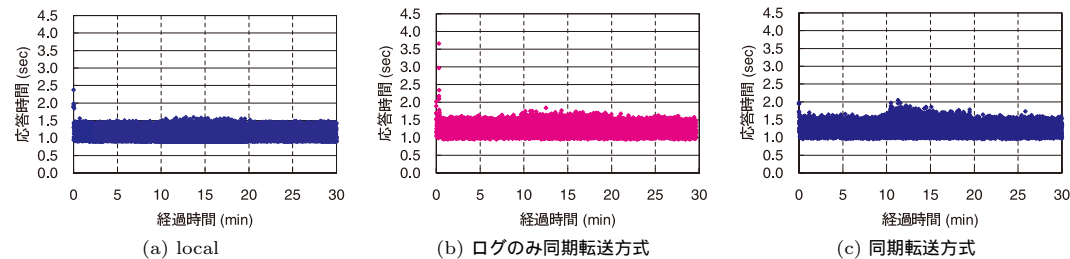


図 20 性能安定性の評価 4 (入力負荷が変動するケース、集中時: ベース負荷 $\times$ 4)

Fig. 20 Evaluation of performance stability3 (fluctuation of work load, peak load: base load $\times$ 4).

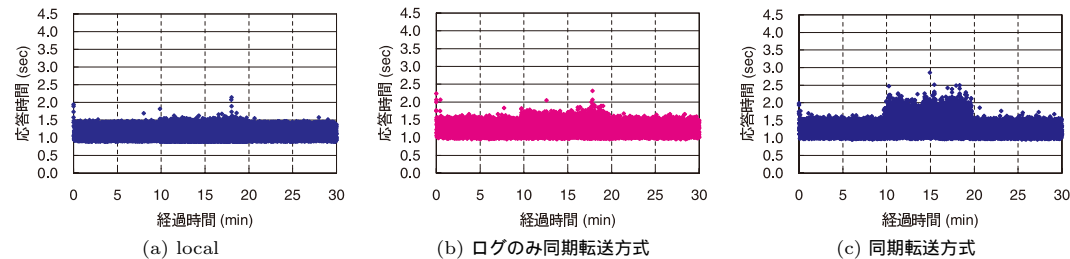
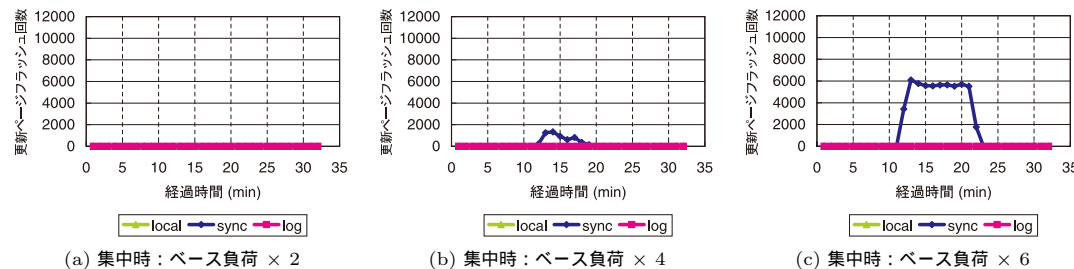
図 21 性能安定性の評価 5 (入力負荷が変動するケース, 集中時: ベース負荷 $\times$ 6)Fig. 21 Evaluation of performance stability³ (fluctuation of work load, peak load: base load $\times$ 6).

図 22 更新ページフラッシュ発生回数

Fig. 22 Number of update page flushes.

3 秒にせまる場合も見られた。

それぞれの負荷スケジュールのもとでの、各 DR システムの更新ページフラッシュ発生回数を図 22 に示す。図 22 (a) より、負荷 1 では各 DR システムとも更新ページフラッシュは生じていない。これは、同期転送方式 (sync) についても、ベースの 2 倍の負荷において、チューニングを行っており、更新ページの出力が間にあっているためである。一方、図 22 (b), (c) に示すとおり、負荷 2、負荷 3 のもとでは、同期転送方式については、更新ページフラッシュが発生している。これは、負荷 2/負荷 3 のもとでは、負荷集中期間には想定していた負荷 (ベース) の 4~6 倍の負荷がかかったため、更新ページの出力が間に合わなくなったためである。その結果、更新ページフラッシュが発生するようになり、図 20、図 21 に示すように応答時間が悪化する。

以上の結果から、ログのみ同期転送方式は、同期転送方式に比べて入力負荷の変動に対し

ても頑強であり、高い性能安定性を示すことが確かめられた。

6. ま と め

本論文では、広域災害対応 DR システムとして同期リモートコピーを利用した 2 つの方式 (ログのみ同期転送方式と同期転送方式) の性能安定性の評価を行った。評価においては、最初に、転送遅延がログ出力と DB の更新ページの出力に与える影響を考慮して、それぞれの方式の性能モデルを構築した。実測との比較を行った結果、性能モデルによって約 1 割の誤差で DR システムの性能を予測可能なことが確かめられた。

この性能モデルから、両方式には性能安定性に差異があることが考察された。そのため、回線シミュレータを用いた DR 評価環境を構築し、転送遅延といった環境パラメータに変動を与えた場合の両方式のオンライン性能を調査することで性能安定性を評価した。評価

の結果、ログのみ同期転送方式は、DBMS 内部の挙動は DR を行わない local 構成と同等であるため、local 構成でのチューニング方法が利用できること、および、転送遅延など環境変動にも頑強で高い性能安定性を示すことが確かめられた。一方、同期転送方式は、DB Writer 多重度が重要なパラメータとなっており、その値が不足、あるいは、過多になった場合に性能低下がみられた。すなわち、同期転送方式では、ある環境に対して最適な DB Writer 多重度が決まってしまうため、環境変動があった場合には、変動前のチューニングが有効でなくなり、新たにチューニングをやり直す必要がある。したがって、同期転送方式は、環境変動の度に DR 向けのチューニングをやり直す必要があり性能安定性が低いのにに対し、ログのみ同期転送方式は、local 環境でのチューニングが利用でき、環境変動にも頑強であり高い性能安定性を示すことが確かめられた。

謝辞 本研究の成果の一部は、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」の支援により得られたものである。

参 考 文 献

- 1) Eagle Rock Alliance: Contingency Planning Research (1996).
- 2) Patterson, D.A.: A Simple Way to Estimate the Cost of Downtime, *LISA Conference* (2002).
- 3) *Disaster Recovery Journal*, Fall (2001).
- 4) 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). <http://www.sec.gov/news/studies/34-47638.htm>
- 5) 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 (2005). <http://www.meti.go.jp/report/data/g50331dj.html>
- 6) British Standards Institution: BS25999: Business Continuity Management (2006).
- 7) British Standards Institution: PAS56: Guide to Business Continuity Management (2003).
- 8) National Fire Protection Association: NFPA1600: Standard on Disaster/Emergency Management and Business Continuity Programs (2004 Edition) (2004).
- 9) Polyzois, C.A. and Garcia-Molina, H.: Evaluation of Remote Backup Algorithms for Transaction-Processing Systems, *ACM Trans. Database Systems*, Vol.19, No.3, pp.423-449 (1994).
- 10) Garcia-Molina, H. and Polyzois, C.A.: Issues in disaster recovery, *IEEE Compcon. IEEE*, New York, pp.222-230 (1990).
- 11) Gracia-Molina, H. and Polyzois, C.A.: Two Epoch Algorithms for Disaster Recovery, *Proc. 16th VLDB Conference Brisbane, Australia*, pp.222-230 (1990).
- 12) King, R.P., Halim, N., Gracia-Molina, H. and Polyzois, C.A.: Overview of Disaster Recovery for Transaction Processing Systems, *IEEE 10th ICDCS. IEEE*, New York, pp.286-293 (1990).
- 13) EMC Corp.: Symmetrix Remote Data Facility product description guide (2000).
- 14) EMC Corp.: EMC Enterprise SRDF consistency group: Description and usage validation, Engineering White Paper (2002).
- 15) Mikkelsen, C. and Attanse, T.: Addressing Federal Government Disaster Recovery Requirements with Hitachi Freedom Storage, White Paper, Hitachi Ltd. (2002).
- 16) 越後, 湯瀬, 干川, 沢野, 高畑, 柴田: 大規模分散環境におけるロバストネスを考慮した広域災害情報共有システム, 情報処理学会論文誌, Vol.48, No.7, pp.2340-2349 (2007).
- 17) 越後, 湯瀬, 干川, 沢野, 高畑, 柴田: 遠隔地ミラーリングを考慮した災害情報ネットワークシステム, 情報処理学会研究報告マルチメディア通信と分散処理研究会報告, Vol.2005, No.58(20050602), pp.81-86 (2005).
- 18) Oracle: DataGuard Overview. <http://www.oracle.com/technology/deploy/availability/htdocs/DataGuardOverview.html>
- 19) 日立製作所: HiRDB DataReplicator Version6 マニュアル.
- 20) No Data Loss with Oracle8i Data Guard and SRDF. http://otn.oracle.com/deploy/availability/pdf/EMC_NOL8i_DataGuard.pdf
- 21) DB2 UDB V7.2, EMC TimeFinder, および EMC SRDF によるデータ・ロスのないスタンバイ・データベースの作成. <http://www-6.ibm.com/jp/software/data/developer/library/techdoc/pdf/standbydb.pdf>
- 22) 山口, 荒木, 後藤: Symfoware ディザスタリカバリ, *FUJITSU*, Vol.57, No.2, pp.126-133 (2006).
- 23) 日立製作所: ログのみ同期転送で通信コストを削減する高信頼ディザスタリカバリ技術, はいたつく, Vol.8, pp.15-16 (2005).
- 24) 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平成 16 年度報告書, pp.79-81 (2005).
- 25) ジム・グレイほか: トランザクション処理—概念と技法, 日経 BP 社 (2001).
- 26) Transaction Processing Performance Council: TPCBenchmark C, Standard Specification, Revision 5.1 (Dec. 2002).
- 27) 日立製作所: Hitachi TrueCopy Basic ユーザーズガイド.

(平成 19 年 12 月 20 日受付)

(平成 20 年 4 月 8 日採録)

(担当編集委員 大森 匡)



鈴木 芳生 (正会員)

1992 年筑波大学第三学群情報学類卒業。1994 年同大学大学院理工学研究科修士課程修了。同年 (株) 日立製作所入社。中央研究所にて可視化システム、データベースシステム等の研究開発に従事。現在、同研究所主任研究員。



渡辺 聡

1997 年早稲田大学理工学部卒業。1999 年同大学大学院理工学研究科修士課程修了。同年 (株) 日立製作所入社。中央研究所にて、性能シミュレーション、データベースシステム等の研究開発に従事。現在、同研究所研究員。



水野 和彦

1992 年熊本工業高等学校電子科卒業。同年 (株) 日立製作所入社。中央研究所にて Grid、データベース等の研究開発に従事。現在、同研究所研究員。



藤原 真二 (正会員)

1988 年京都大学工学部卒業。1990 年同大学大学院修士課程修了。同年 (株) 日立製作所入社。中央研究所に勤務。1998～1999 年スタンフォード大学客員研究員。1999～2001 年日立アメリカ研究員。現在、日立製作所中央研究所主任研究員。データベース、実世界情報処理システムの研究開発に従事。電子情報通信学会、IEEE、ACM 各会員。



河村 信男 (正会員)

1981 年愛媛県立松山工業高等学校卒業。同年 (株) 日立製作所入社。ソフトウェア事業部にてデータベースシステムの開発に従事。現在同事業部主管技師。



喜連川 優 (フェロー)

1978 年東京大学工学部電子工学科卒業。1983 年同大学大学院工学系研究科情報工学専攻博士課程修了。工学博士。同年同大学生産技術研究所第三部講師。現在、同教授。2003 年より同所戦略情報融合国際研究センター長。データベース工学、並列処理、Web マイニングに関する研究に従事。1997 年、1998 年電子情報通信学会データ工学研究専門委員会委員長、1999～2002 年 ACM SIGMOD Japan Chapter Chair、2002 年、2003 年本会理事、2003 年本会フェロー。日本データベース学会理事、SNIA-J 顧問。IEEE TCDE Asian Coordinator、ACM SIGMOD Advisory Board Member。