6D-3

ハードウェア分散共有メモリを用いた広域分散予備方式の評価

横山 和俊　　箱守 聰
(株) NTT データ　技術開発本部　情報科学研究所

1．はじめに

大規模なトランザクションシステムは、サービスダウンによる社会的・経済的な影響が大きく、障害時にも迅速なサービス再開が求められる。筆者らは、ハードウェア分散共有メモリを用いた広域分散予備方式の研究を行っている[1]。提案する方式では、稼動計算機上のアプリケーションの実行状態は、DSM を介して待機系計算機に送信される。稼動計算機に障害が発生した場合には、待機計算機が、メモリレベルで処理を引き継ぎ、高速なシステム切替えが可能にする。本稿では、分散共有メモリを介したトランザクション処理の再開時間の評価について述べる。

2．分散共有メモリを用いた広域分散予備方式

図1に DSM を用いた広域分散予備方式のプロトタイプシステム構成を示す。プロトタイプシステムは、市販の PC（Pentium Ⅲ:600MHz, PCI バス:32bit, 33MHz）を用い、OS は Linux を搭載している。DSM は PCI バスに接続し、Linux の mmap によりアプリケーションの論理空間にマップしアクセスする。DSM は、市販のメモリと FPGA を組み合わせて容易に実現することができ[2]、約 500nsec/1word でアクセスすることが可能である。稼動計算機上で更新された内容は、ネットワークを介して待機計算機の DSM に反映される。

3．トランザクション処理への適用

アプリケーションモデルを図2に示す。端末から送られるメッセージを受付キューに登録（エンキュー）する。振分け処理では、受付キューからメッセージを読出した後（デキュー）、メッセージに応じた振分け処理を実行し、送信キューへエンキューする。受け付けたメッセージは、欠落／重複することなくホスト計算機へ送信する必要がある。

Evaluation of Memory Level Back-up using Hardware Distributed Shared Memory
Kazutoshi Yokoyama and Satoshi Hakomori
Laboratory for Information Technology, NTT Data Corporation
1-21-2, Shinkawa, chuo-ku, Tokyo, 104-0033,Japan

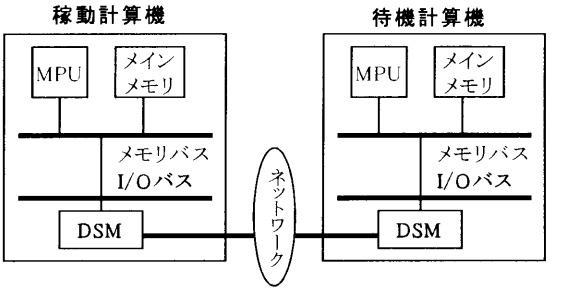

図1　システム構成

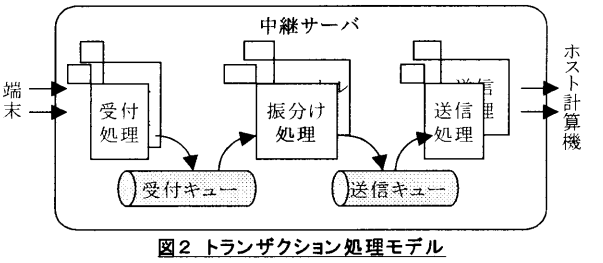

図2　トランザクション処理モデル

振分け処理を例に、処理の詳細を説明する（図3）。DSM は、実際のデータが格納されるデータ領域と領域の状態を管理する管理テーブルから構成される。DSM 上の複数のデータ領域（例えば、A と β）を更新する場合、以下の手順で行う。

(a) A の管理テーブルをデキュー中（ReadDirty）に設定する。
(b) A のデータ領域を更新する。
(c) β の管理テーブルをエンキュー中（WriteDirty）に設定する。
(d) β のデータ領域を更新する。
(e) A と β のデータ更新をコミットする。具体的には、A の管理テーブルを削除済み（Clean）に、

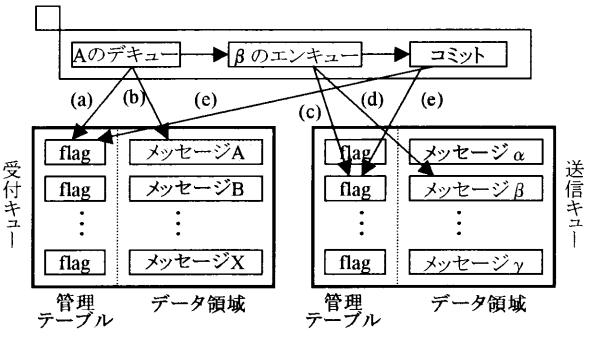

図3　トランザクションの保証

βの管理テーブルを更新済み（Commit）に設定する。

手順 e を実行する場合、DSM のアトミック更新機能[1]により、A とβの管理テーブルが両方更新される、両方更新されないかが保証している。

（2）障害時のサービス再開処理

稼動計算機で障害が発生した場合、待機計算機では DSM の管理テーブルを検索し、ReadDirty ならば Commit 状態にロールバックし、WriteDirty であれば Clean 状態へロールバックする。その後、端末から新しいメッセージを受け付けることができる。

4．障害復旧時間の評価

4．1 管理テーブル数の見積もり

AP の処理が十分に小さいと仮定した場合、DSM アクセスが処理のボトルネックになる。DSM アクセス時間から処理全体の平均スループット値（T）の算出は以下の式で計算できる。

$$T = \frac{1}{\text{ﾒｯｾｰｼﾞｻｲｽﾞ（ﾜｰﾄﾞ数）} \times 500(\text{nsec})} \quad \text{（式1）}$$

例えば、メッセージサイズが 10KByte のとき約 195（メッセージ／sec）となる。

一方、DSM の管理テーブル数は、システムが仮定する最大スループット値によって決定できる。受付処理だけに着目した場合、平均スループット値の 4 倍のメッセージを受け付けることができる。このため、DSM 上の管理テーブルは、受付キューと送信キューを合わせて、平均スループットの 8 倍の個数が必要である。図 4 にメッセージサイズを可変にした場合の平均スループット値、および、必要な管理テーブル数を示す。

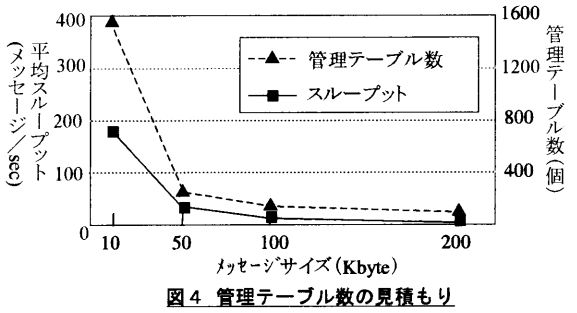

図4 管理テーブル数の見積もり

4．2 ロールバック時間の見積もり

ロールバック処理時の DSM アクセス時間の見積もりを図 4 に示す。ロールバック処理にかかる時間は管理テーブルの個数と管理テーブルの flag の状態に影響される。管理テーブル数を 1600 として仮定した場合、すべての管理テーブルが ReadDirty か

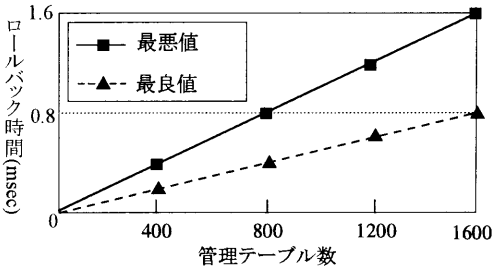

図5 ロールバック処理時間

WriteDirty であれば、最悪 1msec の時間がかかる。

4．3 考察

稼動計算機に障害が発生した場合、待機計算機は、端末側に障害を隠蔽することが理想的である。すなわち、待機計算機でのロールバック処理時間が、端末からのメッセージ到着間隔より小さければ、障害の隠蔽が可能である。図 6 にメッセージ到着間隔とロールバック処理時間の関係を示す。

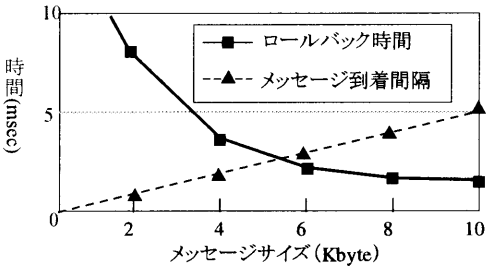

図6 メッセージ到着間隔とロールバック処理時間

図より、メッセージサイズが約 5.5KByte 以上のとき（スループットが約 350 メッセージ／sec 以下のとき）、ロールバック処理がメッセージ到着間隔より小さくできることが分かる。すなわち、提案する方式は、メッセージサイズが大きいアプリケーションに対して、障害を完全に隠蔽することが可能である。

5．おわりに

本稿では、ハードウェア分散共有メモリを用いた広域分散予備方式の評価について述べた。提案方式では、数 msec の高速なサービス再開処理を実現している。また、メッセージサイズが大きいアプリケーションに適している。今後、実測によるサービス再開処理時間の評価を行う。

参考文献

[1] 横山 他：” ハードウェア共有メモリを用いた広域分散予備方式の検討”, 情報処理学会第 60 回全国大会 2J-05 (2000).

[2] 向井 他：” 柔軟な分散共有メモリボードの実現と広域分散予備方式への応用”, 情報処理学会第 60 回全国大会 2J-06 (2000).