高信頼化ミドルウェア ARTEMIS の分散レプリケーション方式

6C－8　Advanced Reliable disTributed Environment MIddleware System

(http://www2.toshiba.co.jp/ilab/artemis)

平山 秀昭[1]　白木原 敏雄[2]　金井 達徳[2]　佐藤 記代子[2]

[1](株) 東芝　情報・通信システム技術研究所
[2](株) 東芝　研究開発センター　情報・通信システム研究所

1　はじめに

分散システム全体の高信頼化を目的に、高信頼化ミドルウェア ARTEMIS (Advanced Reliable disTributed Environment MIddleware System) [1][2] を開発した。本稿では、ARTEMIS の分散レプリケーション方式について報告する。

2　HA システムと ARTEMIS

一般に、分散システムの中でも、サーバコンピュータはクライアントコンピュータに比べて、より高い信頼性を必要とする。これは、1台のサーバコンピュータがダウンすると、そのサービスを受けている全てのクライアントコンピュータが、影響を受けるからである。

サーバコンピュータの信頼性を上げるには、通常まず2重化ディスクや RAID を導入することにより、データの保全性を高める。更に信頼性を上げるには、プライマリコンピュータとバックアップコンピュータによって2重化された HA(High Availability) システムを導入することにより、システム全体の MTTR を改善しアベイラビリティを高める。それよりも更に信頼性を上げるには、MTBF を改善する必要がある。この手段として、HA システムに ARTEMIS を組み合わせることを提案する。

ARTEMIS を組み合わせることにより、HA システムは無停止システム (フォールトトレラントシステム) の領域まで信頼性を高めることができる。　以下では、サーバコンピュータを、プライマリ系とバックアップ系で二重化した HA システム上の、ARTEMIS について説明する。

3　チェックポイントの分散レプリケーション

プライマリコンピュータがダウンしても、そこで実行中だったプロセスを、バックアップコンピュータ上で、継続処理するためには、チェックポイントを、バックアップコンピュータに転送する必要がある。そしてチェックポイントの転送中に障害が発生しても構わないように、チェックポイントは少なくとも2世代保持する必要がある。これにより、任意の時点でプライマリコンピュータがダウンしても、システムをチェックポイントまでロールバックすることができるようになる。

またプロセスが異常終了した場合にも、チェックポイントから再実行することにより、回復を試みる。これは異常終了の原因となったソフトウェアバグが、トランジェントな性質を持つ場合に有効である。このために ARTEMIS では、プライマリとバックアップの両系で、チェックポイントを保持する。プロセス異常終了の場合には、まずプライマリコンピュータ上でチェックポイントから再実行する。回復に失敗するとバックアップコンピュータ上で再実行する。それでも駄目な場合は回復を諦める。

4　共有メモリの分散レプリケーション

データベース管理システムの様な、大規模並列プログラムは、一般に広大な共有メモリセグメントを使用することが多い。これをチェックポイントの度に、全て保存するのでは、オーバヘッドが大きい。

ARTEMIS は、共有メモリをアタッチ / デタッチする時に、その情報を保存する。保存した情報は、次のチェックポイントの一部となる。また、共有メモリ上のデータに関しては、チェックポイント採取時に、直前のチェックポイント以降に更新されたページのみを、プライマリ系に保存すると共に、バックアップ系に転送することにより、オーバヘッドを低減する。

5　ファイルの分散レプリケーション

ファイルは、アドレス空間、プロセッサコンテクスト、セマフォや共有メモリ等の他の OS サービスと、異なる性質を持つ。大半のサービスに関する情報は、チェックポイントとして記録する。そして障害発生時には、全てチェックポイント情報から復元する。しかしファイルに関する全ての状態を、チェックポイントとして保存することには、規模的に無理があるため、図1に示した様な方法を提案する。

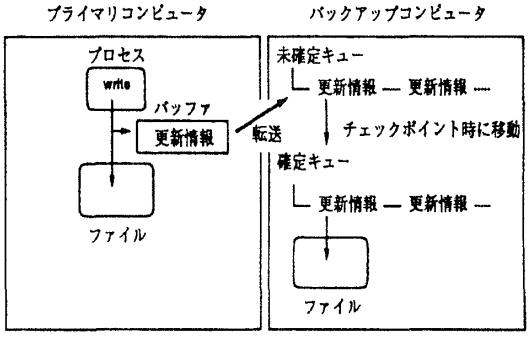

図1：ファイルの分散レプリケーション

Distributed Replication Mechanism of ARTEMIS (Advanced Reliable disTributed Environment MIddleware System)
Hideaki HIRAYAMA:
Information & Communication Systems Laboratory,
TOSHIBA Corporation
Toshio SHIRAKIHARA,Tatsunori KANAI,Kiyoko SATO:
Communication and Information Systems Research Laboratories,
Research and Development Center,
TOSHIBA Corporation

すなわちファイルは、プライマリ系とバックアップ系とで2重化する。プロセスがプライマリ系のファイルを更新する場合には、その更新情報(書き込み位置、書き込みサイズ、書き込みデータ)を記録する。この更新情報はプライマリ系でバッファリングしながら、順次、バックアップ系に転送する。これらの更新情報は、次のチェックポイント時点までに、転送が完了していればよい。

バックアップ系には、未確定キューと確定キューの、2つのキューがある。プライマリ系から転送されてきた更新情報は、一旦、未確定キューにリンクする。次のチェックポイントを経過するまでは、これらの更新情報はそのままにしておく。未確定キューにリンクされた更新情報は、チェックポイント時に、確定キューに移動する。確定キューに移動した更新情報は、順次バックアップ系のファイルに反映していく。

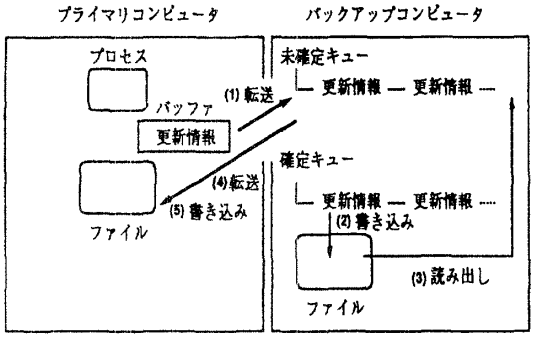

図2: プライマリコンピュータ上のファイルの復元

プロセス障害が発生し、プライマリ系でプロセスを再実行する場合には、プライマリ系で更新中のファイルを、チェックポイント時点の状態に復元しなければならない。図2にこの方法を示す。チェックポイント以降の、プライマリ系のファイルの更新情報は、プライマリ系にバッファリングされているか、バックアップ系の未確定キューにリンクされている。そこで、

1. まずプライマリ系にバッファリングされている更新情報を、バックアップ系に転送し、
2. 次にバックアップ系の確定キューにリンクされている更新情報を、ファイルに反映させ、
3. バックアップ系の未確定キューにリンクされている更新情報の、ファイルの書き込み位置と書き込みサイズに基づいて、その部分のデータをバックアップ系のファイルから読み出し、
4. 最後にそれをプライマリ系に転送し、
5. プライマリ系のファイルに書き込む。

これにより、プライマリ系のファイルを、チェックポイント時点の状態に復元できる。

一方、プライマリ系でプロセス障害が再発したり、プライマリコンピュータがダウンした場合には、バックアップ系でプロセスを再実行する。この場合のファイルの復元方法を図3に示す。これは、

1. 未確定キューにリンクされている更新情報を廃棄し、
2. 確定キューにリンクされている更新情報をファイルに反映させる。

これにより、バックアップ系のファイルを、チェックポイント時点の状態にできる。

また、バックアップ系がダウンした場合には、それをそのまま切り離してしまえばよい。

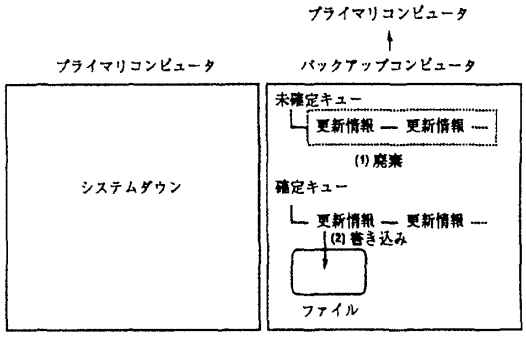

図3: バックアップコンピュータ上のファイルの復元

6 システムの再構成

プライマリコンピュータに障害が発生し、バックアップコンピュータで処理を引き継ぐと、それが新たなプライマリコンピュータとなる。障害を発生したコンピュータがリブートできた場合には、それを新しいバックアップコンピュータとして、HAシステムを再構成することが望ましい。

ARTEMISでは、障害によりシステムが片系で稼働するようになると、チェックポイントの採取を中止する。ただし、システムを再構成した後での、チェックポイント採取再開に備え、OSサービスに関する情報を保存し続ける。ただしファイルの更新情報は膨大になるため、記録しない。

バックアップコンピュータが使用可能になると、再構成を行う。再構成では、

1. 中止していたファイルの更新情報の記録を再開し、
2. オープン中のファイルやクローズされたファイル等を、プライマリコンピュータからバックアップコンピュータにコピーし、
3. それが終わると、チェックポイントの採取を再開する。

これにより、再構成を完了する。

7 試作

UNIX[1]上で、ARTEMISを試作した。ARTEMISの制御の下、HA構成のサーバコンピュータで、データベース管理システムとWWWサーバを、クライアントコンピュータで、WWWブラウザを実行した。WWWブラウザからWWWサーバに処理を依頼し、そこから起動されるCGIアプリケーションが、データベースを更新中に、プライマリ系のサーバコンピュータをダウンさせても、バックアップ系で引き継がれ、処理が正常に終了することを確認した。

8 おわりに

分散システム全体を高信頼化するミドルウェアARTEMISの分散レプリケーション方式について報告した。HAシステムとARTEMISを組み合わせることにより、HAシステムを無停止システムの領域まで信頼性を高めることができる。今後は、より柔軟な構成を採れるようにしていく。

参考文献

[1] 白木原他, "高信頼化ミドルウェアARTEMISの概要とチェックポイント生成方式", 情報処理学会第54回全国大会, 1997年3月.

[2] 佐藤他, "高信頼化ミドルウェアARTEMISの分散チェックポイント生成方式", 情報処理学会第54回全国大会, 1997年3月.

[1] UNIXはX/Openの商標です.