高可用性(HA)の新技術——無停止分散システム構築のためのミドルウェア

New Technology for High-Availability——Middleware for Developing Fault-Tolerant Distributed System

白木原　敏雄　T. Shirakihara
平山　秀昭　H. Hirayama

当社では，高可用性（HA：High Availability）の新技術として，無停止分散システムを構築するためのミドルウェア“ARTEMIS”（Advanced Reliable disTributed Environment Middleware System）を研究開発している。ARTEMISをHAシステムと組み合わせることにより，分散システムを無停止化する。既存のアプリケーションやオペレーティング システム（OS）には，いっさい手を加える必要がない。無停止分散システムを実現するために，分散システム上のプロセスのチェックポイントを採る。サーバ計算機は運用系と待機系で二重化されていて，運用系サーバ計算機で採ったチェックポイントを待機系サーバ計算機に送る。これにより運用系サーバ計算機がダウンしても，待機系サーバ計算機でチェックポイントから処理を継続できる。

As a new high-availability (HA) technology, Toshiba has been conducting research on a new middleware called “ARTEMIS” (advanced reliable distributed environment middleware system) for developing a fault-tolerant distributed system.

The combination of an HA system and ARTEMIS offers fault tolerance in a distributed system. It is not necessary to change existing application programs nor the operating system. Checkpoints for the processes in the distributed system are established to realize a fault-tolerant distributed system. The server computer is replicated as primary and backup systems, with the checkpoints established in the primary server computer also being sent to the backup server computer. In the event of a system breakdown involving the primary server computer, the processes are taken over by the backup server computer.

1 まえがき

近年，情報システムはオープンシステムをベースとした分散システムへと移行している。しかし，一般にこのようなシステムは，従来のメインフレームをベースとした集中システムに比べると信頼性が低いと言われている。

この課題を克服するため，当社ではHA技術の研究開発に取り組んでいる。HAシステムにより，サーバ計算機がダウンした場合でも，そこで行われていた処理を他のサーバ計算機で継続することができるようになる。しかし，データベース管理システム（DBMS）のようなデータの一貫性を保証するシステムでは，処理を他の計算機で継続する前に，ジャーナル リカバリと呼ばれる数分から数十分を要する処理を行う必要がある。

この数分から数十分のシステム停止時間が許容できない業務もある。このような場面を想定し，HAシステムの信頼性をさらに高めるためのミドルウェア“ARTEMIS”の研究開発にも取り組んでいる。HAシステムとARTEMISを組み合わせることにより，無停止分散システムが構築できる。すなわち，サーバ計算機がダウンした場合でも，DBMSはジャーナル リカバリの処理を行うことなく，他の計算機で処理を継続できる。これにより，数分から数十分のシステム停止時間を数秒から数十秒に短縮できる。

図1に，HAシステムとARTEMISの適用システムの例を示す。サーバ計算機は運用系と待機系で二重化されている。このシステムは，World Wide Web（WWW）ブラウザ，WWWサーバ，DBMSを組み合わせたシステムで，クライアント計算機上のWWWブラウザからサーバ計算機上のWWWサーバを介してDBMSにアクセスする。このシステムにHAシステムとARTEMISを組み合わせた場合，運用系サーバ計算機がダウンしてもそこで実行中であったWWWサーバとDBMSは，待機系サーバ計算機で処理を継続できる。この際，DBMSはジャーナル リカバリの処理を行う必要がない。また，WWWブラウザとWWWサーバの間の通信セッションも切れない。

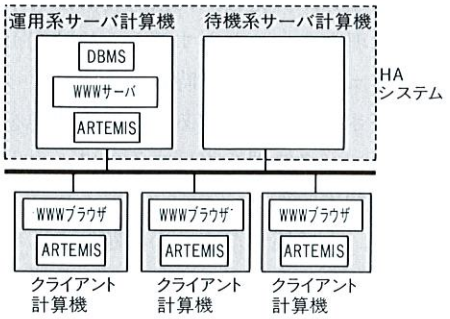

図1．HAシステムとARTEMISの適用システム例　WWWブラウザからWWWサーバを介してDBMSにアクセスする。

Example of HA system with ARTEMIS

2 システム概要

図2に，ARTEMISを導入した場合のシステム全体の処理のようすを示す。このような環境で，ARTEMISは各プロセスのチェックポイントを定期的に採る。この採りかたには次の二つの特長がある。

(1) 各計算機上のプロセスが一斉にチェックポイントを採る。

(2) 運用系サーバ計算機上で採ったチェックポイントを待機系サーバ計算機に送る。

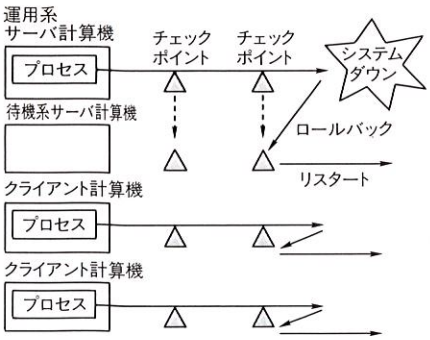

図2．ARTEMIS導入システムの概要　運用系サーバ計算機がダウンしても待機系で処理を継続する。
Overview of fault recovery

図2に示すようなタイミングで運用系サーバ計算機がダウンした場合，運用系サーバ計算機上のプロセスは待機系サーバ計算機上で，クライアント計算機上のプロセスは同じクライアント計算機上で，直前のチェックポイントから処理をリスタートする。ARTEMISは，数秒間隔でチェックポイントを採っているので，障害が発生しても数秒前の状態から処理がリスタートする。途中状態から処理を継続するため，DBMSはジャーナル リカバリの処理を行う必要がない。また，WWWサーバとWWWブラウザの間の通信セッションも切れることがない。

ARTEMISは，特殊なハードウェアの追加，アプリケーションやOSの手直しが不要で，既存システムへのアドオンで分散システムを無停止化する。

3章以降に，ARTEMISを構成する三つのキー技術である“分散チェックポイント”，“分散レプリケーション”，“ジャケット ルーチン”について説明する。

3 分散チェックポイント技術

各プロセスのチェックポイントを一斉に採ろうとしても，分散システムではタイミングのずれが生じ，障害発生時にプロセスをチェックポイントから正しくリスタートできない場合がある。これをメッセージ送受信を行う二つのプロセスA，Bを例に考えてみる。プロセスAはメッセージを送信した後でチェックポイントが採れ，プロセスBはメッセージを受信する前にチェックポイントが採れたとする。障害発生時に，このチェックポイントからプロセスをリスタートすると，プロセスBはメッセージを受信しない（メッセージ喪失）。また，プロセスAはメッセージを送信する前にチェックポイントが採れ，プロセスBはメッセージを受信した後にチェックポイントが採れたとする。障害発生時に，このチェックポイントからプロセスをリスタートすると，プロセスBはメッセージを再度受信してしまう（メッセージ重複）。

図3に，分散チェックポイント技術によるチェックポイント プロトコルを示す。このプロトコルは2相のプロトコルになっており，1相目でプロセスのメッセージ送信を禁止し，2相目でプロセスのチェックポイントを採る。1相目でメッセージ送信を禁止することにより，メッセージ重複のような問題はなくなる。また，メッセージ喪失を回避するために，1相目で各プロセス間のメッセージ量の情報を交換し，送信されたメッセージ量を受信するまで，チェックポイントを採るのを待たせる。このような処理により，分散システム全体で矛盾なくチェックポイントを採ることが可能になる。

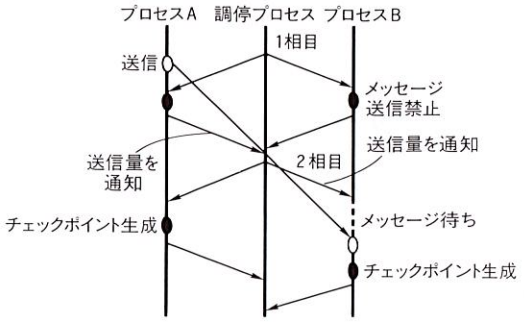

図3．分散チェックポイント プロトコル　分散システム上のプロセスが一斉にチェックポイントを採る。
Protocol of distributed checkpoint

4 分散レプリケーション技術

運用系サーバ計算機に障害が発生した場合，待機系サーバ計算機で処理を継続する。そのために，運用系サーバ計算機上で採れたチェックポイントを待機系サーバ計算機にも送っている。運用系サーバ計算機上で実行中だったプロセスを，待機系サーバ計算機上でチェックポイントからリスタートするためには，運用系サーバ計算機上で行われたファイルの更新が待機系サーバ計算機上にも反映されてい

る必要がある。また，共有メモリについても同様のことが言える。

次に，二重化されたサーバ計算機の間でファイルなどを複製する分散レプリケーション技術について説明する。

図4に，運用系サーバ計算機と待機系サーバ計算機の間で，ファイルの一貫性を維持するための分散レプリケーションの仕組みを示す。アプリケーションがファイルを更新する際に，更新情報を取得して待機系サーバ計算機に送る。この更新情報は，次のチェックポイント経過後に待機系サーバ計算機のファイルに反映する。このような処理により，障害発生時に，運用系サーバ計算機のファイルを直前のチェックポイントの状態にすることが可能になる。

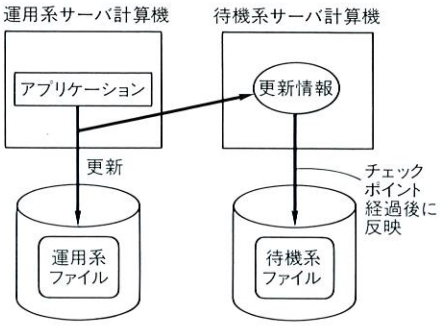

図4．分散レプリケーション機構　運用系サーバ計算機でのファイル更新を待機系にも反映する。
Mechanism of distributed replication

5 ジャケット ルーチン

ジャケット ルーチンは，プロセスが生成される際に動的にリンクされ，システムが提供するインタフェースと同じインタフェースをもつ。そして，プロセスが発行するシステムコールを監視し，そのプロセスのOS内部の状態に関する情報を保存する。これにより，アプリケーションの手直しなしに，チェックポイントを採ることが可能になる。

図5に，通常のプロセスAと，ジャケット ルーチンがリンクされているプロセスBのようすを示す。これらのプロセスは，プログラムで記述された計算などを行うほかに，ファイル操作などOSの提供するサービスを利用するためにシステムコールを発行する。プロセスBにリンクされたジャケット ルーチンは，このシステムコールを監視する。プロセスのチェックポイントは，次の2種類の情報を含む。

(1) アドレス空間およびレジスタセット

(2) OS内部の状態

アドレス空間およびレジスタセットは，ジャケット ルーチンから直接読み書きすることにより，保存する。OS内部の状態は，ジャケット ルーチンでシステムコールを監視することにより，必要な情報を保存する。

(注1) UNIXは，X/Openカンパニーリミテッドがライセンスしている米国ならびに他の国における登録商標。

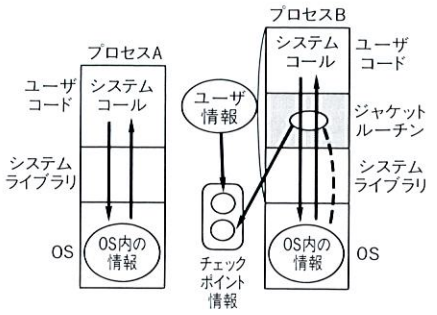

図5．ジャケット ルーチンの仕組み　システムコールを監視してチェックポイントを採る。
Mechanism of jacket routine

6 あとがき

ARTEMISをHAシステムと組み合わせることで，無停止分散システムの実現が可能となる。その際，アプリケーション プログラムの手直しは不要である。ARTEMISは，現在UNIX(注1)上で評価中である。また，他のシステム上への展開も行っている。この技術により，幅広い信頼性への要求にこたえられるよう，今後も研究開発を進めていく。

文 献

(1) 白木原敏雄，他：高信頼化ミドルウェア“ARTEMIS”の設計と実装，情報処理学会 研究報告 97-OS-74，pp.183-188 (1997)

(2) 平山秀昭，他：高信頼化ミドルウェアARTEMISを用いた無停止分散システム，電子情報通信学会 技術研究報告 FTS97-19，pp.21-26 (1997)

(3) T. Shirakihara, et al : ARTEMIS : Advanced Reliable disTributed Environment MIddleware System, Proceedings of PDPTA'97, pp.97-106 (1997)

白木原　敏雄　Toshio Shirakihara

研究開発センター 情報・通信システム研究所研究主務。OS・ミドルウェアでの高信頼性の研究開発に従事。情報処理学会会員。
Communication & Information Systems Research Lab.

平山　秀昭　Hideaki Hirayama

情報・通信システム技術研究所 開発第一担当主務。OS・フォールトトレラントシステムの研究開発に従事。情報処理学会，電子情報通信学会会員。
Information & Communications Systems Lab.