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

6C−6

(Advanced Reliable disTributed Environment MIddleware System)

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

白木原 敏雄[1]　平山 秀昭[2]　佐藤 記代子[1]　金井 達徳[1]

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

1 はじめに

近年、イントラネット上でのデータベース (DB) 処理をクライアント・サーバシステム上で行う等、分散処理が広く使用されるようになっている。このような環境では、サーバ計算機に障害が発生した場合そのサーバ計算機が行っているサービスが使用できなくなるため、信頼性が重要になってくる。高信頼化ミドルウェア ARTEMIS(Advanced Reliable disTributed Environment MIddleware System) は分散システム全体の高信頼化を実現するものであり [1, 2]、基本メカニズムとして、チェックポイント・リスタート方式をベースにしている。本稿では、ARTEMIS の概要およびプロセスのチェックポイント (CP) 生成方法について述べる。

2 ARTEMIS の概要

ARTEMIS はイントラネット上の分散処理、特にデータベースサーバ、Web サーバ、Web ブラウザの 3 層で構成される 3 層クライアント・サーバシステムやグループウェアをターゲットにシステム全体の高信頼性を提供するミドルウェアである。

図 1は、3 層クライアント・サーバシステムの構成例を示したものである。ユーザはクライアント計算機上の Web ブラウザから Web サーバに DB アクセス要求を行う。Web サーバはデータベースシステムをアクセスし、結果を HTML に変換して Web ブラウザに返す。このような環境において処理途中に DB サーバのサーバ計算機に障害が発生した場合、それ以降の処理が継続できなくなる。すなわち、ハードウェアや OS のバグによるダウンの場合、OS がリブートされ、DB のジャーナルリカバリが行われるまでの数十分、致命的

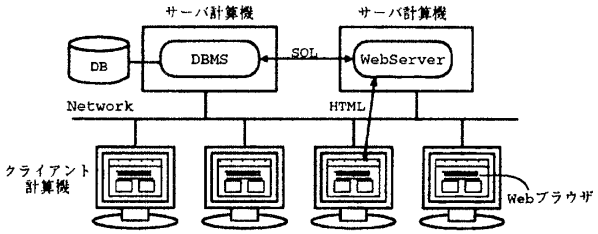

図 1: 3 層クライアント・サーバシステムの例

なハードウェア故障の場合、修理が完了するまで、最悪数日、サービスが停止してしまうことになる。

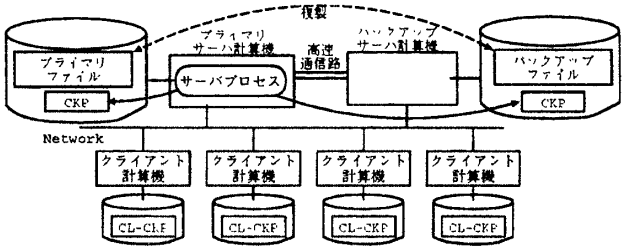

図 2: ARTEMIS の処理概要

図 2は、ARTEMIS の処理概要を示したものである。ARTEMIS はチェックポイント・リスタート方式をベースに、以下の 3 つのキー技術により、上記の問題を解決する。

- 分散処理に対応した分散 CP 生成技術
- 2 重化されたサーバ計算機で、1 台の計算機に障害が発生した場合に他方で即時引継ぎを可能にする分散レプリケーション技術
- プロセス単位の CP をアプリケーションの変更なしに生成するジャケットルーチン

分散 CP 生成技術は分散システム上でプロセス間通信を行う複数のプロセスの CP 生成を矛盾なく生成するためのプロトコルを実現している。

Overview and Checkpointing Mechanism of ARTEMIS (Advanced Reliable disTributed Environment MIddleware System)

Toshio SHIRAKIHARA, Kiyoko SATO, Tatsunori KANAI: Communication and Information Systems Research Laboratories, Research and Development Center, TOSHIBA Corporation

Hideaki HIRAYAMA: Information & Communications Systems Laboratory, TOSHIBA Corporation

分散レプリケーション技術では、プライマリ計算機上のプロセスのCP情報をバックアップ計算機側に転送するとともに、2台のサーバ計算機間で複製ファイルをサポートする。これにより、プライマリ計算機障害時にバックアップ側での引継ぎを可能にする。

ジャケットルーチンは、プロセス起動時に動的にリンクされるダイナミックリンクライブラリで、リンクされたプロセスのCP生成を行う。ジャケットルーチンはOSと同じシステムコールAPIを持ち、実行時に動的にリンクされるため、アプリケーションおよびOSの変更は不要(バイナリコンパチ)である。

以降では、3つのキー技術の1つであるジャケットルーチンについて説明する。

3 チェックポイント生成方式

ジャケットルーチンでは、プロセスが発行するシステムコールをフックし、プロセスの状態の保存(CP生成)・回復(リスタート)を行う。プロセスの状態はOSに依存する部分が多いが、本稿では、UNIX[1](Solaris[2])の場合について述べる。

プロセス資源はアドレス空間およびレジスタセットといったプロセスから直接アクセスできる資源と、ファイル、共有メモリ、socketといったOSにより提供される資源の2つに大別できる。アドレス空間の情報は、前回のCP生成時から現在までに更新されたページ(ダーティページ)だけを保存する差分保存方式をとっている[3]。レジスタセットに関しては、setjmp/longjmpにより保存・回復を行う[4]。OS資源についてはプロセスがOS呼び出しを行うシステムコールをジャケットルーチンでフックし、必要な情報を保存・回復する。

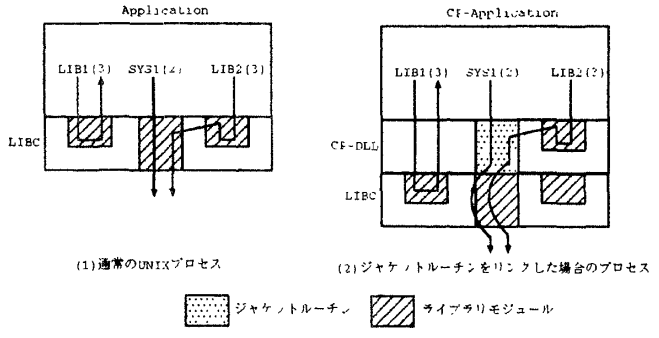

図3: システムコールのフック方法

図3はシステムコールをフックするための仕組みを示した図である。通常のプロセスは図3(1)に示すようにOSをアクセスするためのシステムライブラリ(LIBC)がリンクされる。LIBCは通常、以下の3種類のインタフェースを提供する。

- システムコール(SYS1、ex. open)
- ライブラリコール(LIB1、ex. strcpy)
- 間接的システムコール(LIB2、ex. fopen)

これに対してARTEMISでは、図3(2)に示すようにジャケットルーチンがLIBCより先にリンクされる。ARTEMISでは必要なシステムコールをフックするジャケットルーチンを持っており、アプリケーションがシステムコールSYS1を発行した場合、LIBCのSYS1ではなく、ジャケットルーチンのSYS1が呼ばれる。また、LIB2のように間接的にSYS1を呼び出すような関数に関しては、LIBCからLIB2のオブジェクトを取り出し、ともにリンクする。この場合LIB2からのSYS1の呼び出しはSYS1を呼び出すように解決されるため、最終的にはジャケットルーチンSYS1が呼び出される。

このように、ジャケットルーチンをアプリケーションにシステムライブラリより先にリンクすることにより、アプリケーションの変更を必要とせず、そのアプリケーションに関するOS内の状態を保存することが可能になる。

4 おわりに

本稿では、高信頼化ミドルウェアARTEMISの概要とプロセスのCPを生成する仕組みであるジャケットルーチンについて述べた。アプリケーションに関するOS内の状態の保存・回復方法はこの技術のポイントの1つであるが、その詳細については別の機会で述べたい。

参考文献

[1] 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", 情処第54回全国大会,1997.
[2] 平山他, "高信頼化ミドルウェアARTEMISの分散レプリケーション方式", 情処第54回全国大会,1997.
[3] J. S. Plank, M. Beck, G. Kingsley and K. Li, "Libckpt: Transparent Checkpointing under UNIX", USENIX Winter 1995 Technical Conference, 1995.
[4] 森山他, "利用者レベルで実現したプロセス移送ライブラリ"、情処OS研究会報告 91-OS-51, 1991.

[1]UNIXはX/Openの商標です。
[2]Solarisは米国SunMicrosystems社の商標です。