6C−7

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

(Advanced Reliable disTributed Environment MIddleware System)

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

佐藤 記代子[1]　白木原 敏雄[1]　平山 秀昭[2]　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

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) を開発した [1, 2]。本稿では、ARTEMIS の分散チェックポイント (CP) 生成方式について述べる。

2 分散チェックポイント生成の問題点

通常、分散システム上では、複数の計算機上のプロセスがメッセージ送受信等のプロセス間通信 (IPC) を行ないながら処理を進める。このような環境において、各プロセスがそれぞれ独自のタイミングで CP を生成した場合、矛盾が発生し、リスタートできないケースがでてくる。

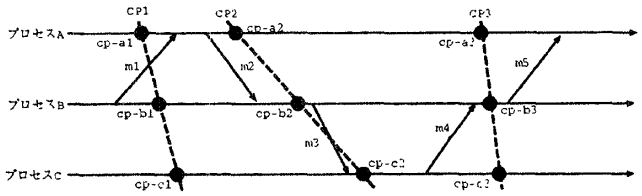

図 1: 分散 CP 生成のタイミング例

図1は3つのプロセスがメッセージ送受信を行なっている場合の CP 生成のタイミングの例を示したものである。図中の CPi はある時点の3つのプロセスの CP のセット、cp-ji は CPi 時点でのプロセス j の CP を示している。それぞれのタイミングにおいて、CP 生成直後、障害が発生してリスタートした場合を考えると以下のようになる。

CP1(NG): プロセス B はメッセージ m1 を送り終わった状態で、プロセス A はメッセージ m1 を受ける前の状態のため、メッセージ m1 が失われてしまう

CP2(NG): プロセス B はメッセージ m3 を送る前の状態で、プロセス C はメッセージ m3 を受けた後の状態のため、メッセージ m3 が 2 回送られることになる

CP3(OK): 矛盾を起こすメッセージがないので、正しくリスタートできる

このように、プロセス毎のタイミングで CP を生成したのでは、正しくリスタートできないケースがでてくる。すなわち、以下のようなメッセージが存在しない CP が一貫性のある CP であるということができる。

- "送信されたが、受信されていないメッセージ"(CP1 の場合)
- "送信されていないが、受信されたメッセージ"(CP2 の場合)

この問題を解決するプロトコルがいくつか提案されているが [3]、それらはメッセージ送受信を対象にしており、その他の IPC(共有メモリ、パイプ、セマフォ等) への対応は困難である。

3 ARTEMIS の分散チェックポイントプロトコル

図 2に ARTEMIS の分散 CP プロトコルを示す。ARTEMIS の環境では、各計算機上に同一計算機上のプロセスを管理するデーモン PM が存在する。また、各プロセスは送信停止フラグをもち、メッセージ送信のジャケットルーチンでは送信停止フラグをチェック

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

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

Hideaki HIRAYAMA: Information & Communications Systems Laboratory, TOSHIBA Corporation

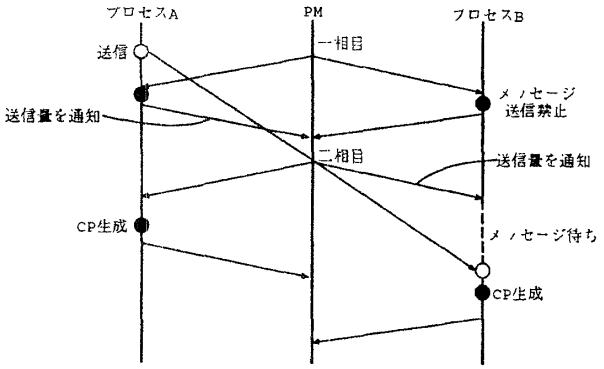

図 2: ARTEMIS の CP プロトコル

し、セットされていなければ送信を行い、セットされていればリセットされるまで送信を待機する。

図は同一計算機上のプロセス A、B がメッセージ送受信を行なっている場合の CP 生成の手順を示している。PM は一相目で同一計算機上のプロセスに以降のメッセージ送信を禁止するように伝える。各プロセスは送信停止フラグをセットし、PM に応答を返す。PM は全ての要求の応答が揃った時点で、二相目を開始する。各プロセスは二相目の要求を受けて、プロセスの CP を生成し、送信停止フラグをリセットする。すべてのプロセスの CP 生成が正常に終了した時点で分散チェックポイントが成立することになる。このように、最初にすべてのプロセスのメッセージ送信を停止した後、CP 生成を行なうことで、"送信されていないが、受信されたメッセージ" が存在しないことになる。また、各プロセスは二相目開始までの間、メッセージ送信以外の処理は継続できるため、分散チェックポイント生成によるプロセスへの影響を小さくできる。

このとき、図に示すように、CP 生成プロトコル開始より前にメッセージを送信して、まだそれを受信していない場合があるため、各プロセスは一相目の応答時に、前回の CP から現在までの間に、プロセス B に送信したメッセージの総量を共に通知し、PM は二相目の開始時にその送信量をプロセス B に通知する。プロセス B は通知された送信量を受信するまで、チェックポイント生成を遅延する。このような処理により、"送信されたが、受信されていないメッセージ" が存在しないことになり、図のようなタイミングでも正しく CP が生成できる。

通信プロトコルの主なものとして、TCP、UDP があるが、送信量の交換を必要とするのは、TCP の場合のみである。すなわち、プロセス A とプロセス B の間での接続に関して、送信量の交換を行なう。これに対して、UDP プロトコルでは、メッセージロストの可能性を許している為、"送信されたが、受信されていないメッセージ" が存在してもよい。そのため、ARTEMIS では、UDP については、送信量の交換および CP 生成の遅延を行なわない。

図 2では、同一計算機上のプロセスの例を説明したが、実際には、複数の計算機にまたがってこのプロトコルが実行される。ARTEMIS の環境では、計算機間の調停を行なうデーモン GM 存在する。GM は一相目で各計算機の PM に一相目の通知をし、全ての PM からの応答がそろったところで、二相目を開始し、すべての PM からの応答がそろったところで、分散 CP が成立する。これにより、図 2と同様の効果が複数の計算機環境で実現できる。

また、これまでの説明では、メッセージ送受信の場合について述べたが、ARTEMIS では、IPC を計算機間 IPC と計算機内 IPC に分類している。計算機間 IPC としては、ネットワークを介したメッセージ送受信があり、計算機内 IPC として、共有メモリ、ファイル共有によるデータ交換、セマフォ、パイプ等がある。計算機間 IPC については、これまで述べた方法で調停を行なう。計算機内 IPC については、CP プロトコルの一相目でアクセスを停止するのではなく、二相目でセマフォ等を利用してプロセス間で同期をとり、同一計算機上のプロセスのアクセスが停止された状態にして CP 生成を行なう。

4 おわりに

本稿では、高信頼化ミドルウェア ARTEMIS の分散 CP 生成方式について述べた。この CP プロトコルは、メッセージ送受信の他に、共有メモリやファイル共有によるデータ交換やセマフォ等の IPC を行なうプロセスについても有効である。ARTEMIS の CP プロトコルは全てのプロセス間で同期して CP 生成を行なう同期型アルゴリズムであるが、今後は非同期型アルゴリズム等を検討し、CP 生成の効率化を図って行きたい。

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] 真鍋他, "分散チェックポイント・ロールバックアルゴリズム", 情報処理,Vol.34,No.11,pp.1366–1467,1996.