HAシステムにおける耐障害処理方式

1B－10

蔵野 政行[1]，広兼 茂和[1]，鈴木 均[1]，末永 司[1]，遠藤 浩太郎[2]
[1](株)東芝 府中工場，[2]（株）東芝 情報・通信システム技術研究所

１．はじめに

ＨＡ(High Availability)システムとは複数台の計算機を通信インタフェースで接続し、相互に計算機の状態を監視することで、システム全体の可用性を向上させることを目的としたシステムである。ＨＡシステムでは、マスタ系の障害をスレーブ系が検出するとスレーブ系によってマスタ系の処理の引き継ぎが行われる。

本報告では、”マスタ系がシステムに複数存在してしまう状態(split-brain syndromeと呼ばれている)を解決する他系監視機構”と”障害検出方式の高速化手法”について述べる。

２．計算機障害と技術課題

ＨＡシステムを構成する系の障害検出は、専用パスおよびＬＡＮにより系間で定期的にメッセージ送受信を行い、他系からのメッセージを監視する方式をとるのが一般的である。他系の障害を検出すると、データの保全性、サービスの継続性を保証するため共有ディスクやＩＰアドレスの健全な系への切り替えを行う。例えば、Ａ系、Ｂ系の２台系のシステムを考えてみる。Ａ系がマスタの時、システム設計値以上の高負荷になった場合メッセージ送信が滞り、Ｂ系により障害発生と検知され、Ｂ系はマスタになる。しかし、Ａ系は障害により停止しているわけではないので、負荷が軽くなると再びマスタとして処理を再開する。こうしてＡ系、Ｂ系がマスタとして動き、split-brain syndromeが発生する。

もう一つの課題は、自系監視での検出時間に関するものである。系間の相互監視はメッセージ送信を使用するため障害を検出するまでの時間が長くなってしまう。そのため自系の障害を自系自身で検出する方法としてウォッチドッグタイマによる自系監視方式が実用化されている。この方式は、監視プロセスが一定時間内にウォッチドッグタイマにアクセスすることを監視しているため、プロセススケジュールの精度、その系の負荷状況に影響されやすいので障害検出時間短縮には限界がある。

３．split-brain syndromeを解決する他系監視機構の実現

split-brain syndromeを解決するための最も効果的な方法は障害が検出された系を確実に停止させることである。障害が検出された系を確実に強制停止することができればsplit-brain syndromeが発生することはない。他系を強制停止する方式をとる上で克服すべき項目は以下の２点である。

(1)健全な系が誤って強制停止要求を受けてしまう点。

(2)一時的な高負荷により２つの系が互いに他系の障害を検出した場合に２台とも強制停止されられてしまう点。

(1)の課題を克服するために、他系からの強制停止要求を自系にてブロックする機能を設けた。

ＨＡシステムに組み込まれていて正常に動作している場合に他系より強制停止要求が行われてもＨＡ制御により他系へ切替処理が行われるので問題とならない。問題となるのはＨＡシステムに組み込まれていないときに他系より強制停止要求が行われることである。自系をＨＡシステムに組み込

A node watching and a failure detection on HA system

Masayuki KURANO[1], Shigekazu HIROKANE[1], Hitoshi SUZUKI[1], Tsukasa SUENAGA[1], Kohtaro ENDOH[2]
[1]Fuchu Works, TOSHIBA Corp.
[2]Information & Communications Systems Lab., TOSHIBA Corp.

む時に他系からの強制停止要求を受理可能（アンブロック）とし、ＨＡシステムに組み込まれていない時には強制停止要求をブロックする方式とした。

(2)の課題を克服するためにも前記ブロック方式を使用し,両系が同時に他系からの強制停止要求をブロックできないよう制御する。他系の障害を検出すると自系が他系より強制停止要求を受理しないようブロックしてから他系へ強制停止要求を行うことにより、２系が互いに他系障害を検出してもどちらか一方は強制停止を回避できる。ＨＡシステムが３台系以上の場合にもこの方法にて対応が可能である。確実性、高速性が重視されるため、高優先度割込により計算機を強制停止させる機能と双方向通信機能をもつ専用の障害検出装置を開発し、各計算機に実装した。（図１）

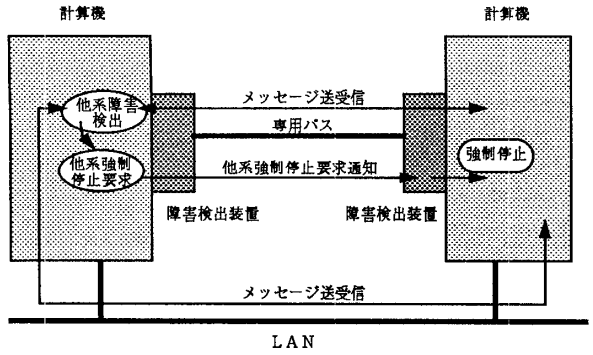

図1．ＨＡシステムにおける他系監視機構

４．障害検出方式の高速化手法

監視プロセスレベルの監視だけだとシステム負荷やスケジューリング精度に影響されるため監視タイムアウト時間を短くできない。また、カーネルドライバレベルの監視だけだとアプリケーションレベルの異常が発生していてもドライバは動作している場合があるためアプリケーションレベル異常の検出ができない。これらの問題を解決するためカーネルドライバによりハードウェアおよびカーネルレベルの異常を検出する機構と監視プロセスによりアプリケーションレベルの異常を検出する機構を組み合わせた障害検出方式を開発した。本方式ではハード的なウォッチドッグタイマー一つで監視が可能である。

＜カーネルドライバによる検出機構＞

カーネルタイマサービスによってカーネルドライバを周期的に動作させ、障害検出装置に設けたタイマを再設定させる。ハードウェアおよびカーネルレベルで異常が発生し、ドライバが動作しなければ障害検出装置によりタイムアウトが検出され、自系を高優先度割込により強制停止し、専用パスにより他系の障害検出装置へ瞬時に通知する。ドライバのカーネルサービスによる周期スケジュールを利用することにより、数ｍｓオーダの監視が可能となる。（図２）

＜監視プロセスによる検出機構＞

ドライバの内部バッファにカウンタを作成し、ドライバが前記周期動作をするタイミングにてカウンタをデクリメントし、アプリケーションプロセスコンテキストで動作する監視プロセスが定周期にカウンタ値を再設定する。アプリケーションレベルの動作異常が発生し監視プロセスの動作が滞るとカウンタが０となり、ドライバはアプリケーションレベルでの動作異常を検出し障害検出装置により自系を強制停止させる。さらに、プライオリティの異なる複数の監視プロセスにカウンタを持たせ、下位プライオリティの監視プロセスが上位プライオリティの監視プロセスのカウンタを再設定する機構を追加することによりプライオリティごとにアプリケーションレベル動作監視の監視周期を設定することが可能となった。（図２）

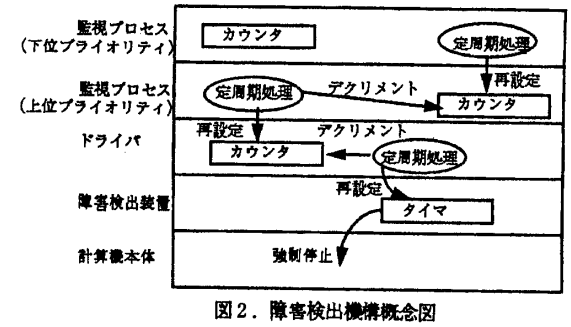

図2．障害検出機構概念図

５．おわりに

split-brain syndromeを解決する他系監視機構と、障害検出方式の高速化手法について述べた。これらはＨＡシステムの高速高信頼障害検出方式として利用できる。