2S－3　大規模高信頼性プラットフォームの構築(3) ―自律分散システム構築のプラットフォーム

坂田 和久†　地引 昌弘†　高田 寛†　三輪 隆弘††

†NEC ネットワーキング技術研究所
†† 日本電気マイコンテクノロジー(株)

1　はじめに

分散環境では、通常、人間の管理者が個々のワークステーションの管理を行なわなければならない。しかし、ワークステーションが、全国規模で展開される広域分散環境においては、管理者の工数が莫大なものになり、全国規模の分散環境は非現実的な解になってしまう。

本稿では、各ワークステーションに管理者が存在していなくても安定した運用を可能にする、自律分散システムについて、その機能と動作を説明する。

2　システム設計上の要件

自律分散システムプラットフォームは、ワークステーションが広範囲に渡って多数存在するようなシステムで起こり得る障害を、自動的に復旧するか、ネットワークを通じて通知を行なう。これにより、管理者が遠隔地にいて、マシンの台数よりも大幅に少ない場合でも、障害の対処が速やかに行なわれることを目的とする。

システム設計上の要件としては次のものがある。

(1) アプリケーションが何らかの障害を検出した場合に、検出した障害の情報を一元的に収集することで、対処もしくは管理者への通知を行なう

(2) システムの状態を常に監視し、なんらかの障害が発生した場合には対処もしくは通知を行なう

3　自律分散システムプラットフォームの構成

自律分散システムは、以下のような構成を取る。

1. RAS[1]ライブラリ,RAS デーモン,RAS 診断

   アプリケーションやミドルウェアの内部で検出されるエラー情報を、一元的に収集し、必要に応じて対処や遠隔地の管理者に通知する

2. システム整合性チェック

   各種システム情報を監視し、ユーザによって設定された状態から逸脱した場合を異常とし、その復旧および管理者への通知を行なう

4　RAS ライブラリ,RAS デーモン,RAS 診断

RAS ライブラリ,RAS デーモン,RAS 診断は、次のような情報の流れを提供する。

(1) エラーを検出したアプリケーションが、エラー情報を RAS 構造体に格納し、RAS ライブラリを呼び出す。

(2) RAS ライブラリは、RAS デーモンに、RAS 構造体の情報を渡す。

(3) RAS デーモンは、RAS 構造体の情報を RAS 情報ファイルを通じて RAS 診断に渡す。

(4) RAS 診断は、RAS 情報ファイルの情報に対して設定された対処を行なう[2]。

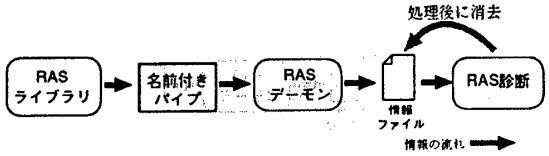

図 1: RAS 情報の流れ

5　システム整合性チェック

システム整合性チェックは、次の項目を監視し、項目毎に対処を行なう。通知を行なう場合には、RAS ライブラリを使用する。

プロセス監視

- 常駐プロセス

  プロセスの存在と CPU 時間、消費メモリを監視し、問題がある場合には停止、再起動を行なう

- 非常駐プロセス

  存在しているプロセスについて、CPU 時間、消費メモリ、存在時間を監視し、設定の範囲を超えている場合には停止する

"A Platform for Highly-reliable Large Scale systems(3) – Autonomous Distributed system platform"
by Kazuhisa Sakata†, Masahiro Jibiki†, Hiroshi Takada†† and Takahiro Miwa††
†NEC Networking Systems Laboratories.
††NEC Microcomputer Technology, LTD.

[1] Reliability Availability Serviceability

[2] 通常の場合は、ネットワークを通じての管理者への通知となる。

ファイル監視

- core

  指定したディレクトリ以下にcoreファイルの存在を調べある場合には通知する

- 必須ファイル

  指定ファイルが存在し、正しいパーミッションとサイズを持つことを確認する。もし違う場合には通知する

ファイルシステム監視

- 容量

  ファイルシステムの残り容量を監視する。もし残り容量が設定値を下回っている場合には通知する

- アクセスのないファイル

  指定ディレクトリ下の長期間アクセスされないファイルを消去する

- fsck

  fsckを自動的に実行し運用中に必然的に出るエラー以外のエラーを検出した場合には通知する

カーネル／ハードウェア情報

カーネルから、システム資源情報またはハードウェアに関する情報を取りだし、設定した範囲から外れている場合には通知する

6 適用結果

実際に展開するシステムは、ワークステーションが業務情報管理システムとして日本全国に数百から数万程度存在し、管理者のいるセンターと、ISDN回線で接続されている。

そのうち164箇所で延べ4151日分に渡って、RASライブラリを使用して通知されたエラーを収集し、次のように分類した。

(1) システム整合性チェックが回復できる障害

バグによる異常終了／暴走	0.9%
動作環境／操作ミスによる異常終了／暴走	5.0%
システムに原因のある異常終了／暴走	3.3%
原因不明の異常終了／暴走	48.3%
計	57.5%

(2) RASライブラリを通じての通知により、遠隔地の管理者が対応することで回復する障害

FSの溢れ	0.0%
fileの消失	0.1%
fileアクセスに対するエラー(permission, サイズ)	10.9%
ハードの故障による障害	1.1%
環境設定(電源入れ忘れ)ミスによる障害	0.6%
計	12.7%

(3) リトライを行なえば正常に復帰する障害

NWを介した通信のタイムアウト	2.5%
NWを介した通信の通信エラー	1.8%
NW回線のビジー	16.6%
システムコールの失敗	7.7%
ローカルIPCのタイムアウト	0.2%
ローカルIPCの通信エラー	0.2%
計	29.0%

(4) 自律分散システムでは回復も通知も行なわれない障害

OSのパニック／ストール	0.8%
計	0.8%

ここで、(1)～(3)までは、システム整合性チェックもしくは遠隔地の管理者により対処されるか、自然に回復する障害であり、その合計は99.2%である。

すなわち、自然に回復もしなければ,管理者に通知されて対処される可能性もない障害は、全体の0.8%に過ぎず、残りは自律分散システムプラットフォームにより、何らかの対処[3]が行なわれるか、自然に解消してしまう障害である。

また、利用者がなんら障害に対するアプローチを行なわずとも自動的に回復する障害である(1)と(3)の割合は、以下に示す通り全体の86.5%である[4]。

よって、自律分散プラットフォームにより、システムは安定性は確保されていると言える。

7 おわりに

運用結果の分析から、決まったアプリケーションしか動作しない業務情報管理システムのようなシステムに対して、自律分散プラットフォームが有効に働くことがわかった。

しかし、同時に以下のような問題点もあることがわかった。

- コンフィグレーションファイルが全国に存在する各マシン毎に用意しなければならなかったため、設定の更新に多大な手間を要した
- 決まったアプリケーションしか動かさないわけではない、一般的なワークステーションの運用には、適用できない機能がある

今後は、柔軟に設定を変更することができ、ある程度の幅を持つ運用がされているマシンの管理も可能なシステムを構築して行きたい。

[3] ここでいう対処には、管理者への通知された結果行なわれたものも含まれる。

[4] (2)は、遠隔地に居る管理者によって利用者に指示が出される形で、利用者がマシンを操作することがある。