

L-075

自律運用管理システムにおける情報更新スケジュール機能 Information Update Scheduling for Autonomous Management Systems

町田 文雄[†] 川戸 正裕[†] 前野 義晴[†]
Fumio MACHIDA Masahiro KAWATO Yoshiharu MAENO

1. はじめに

企業システムの運用管理コストが問題となっていることから、システムの障害や負荷変動に対して自律的に対処する自律運用管理システムが注目されている[1]。自律運用管理システムの実装においては、問題が発生した際に迅速かつ正確に対処することが要求される。このためには制御対象となる各資源の状態を把握する必要があるが、分散した複数の資源情報を収集する処理によって問題対処が遅延してしまう可能性がある。収集した情報をキャッシュして利用することで問題対処の高速化を図れるが、参照可能な情報の鮮度は低下してしまう。本稿では、情報の鮮度低下による自律運用管理システムの対処の遅れや判断ミスを防ぐため、キャッシュの更新処理をスケジュール化して実行する方法を提案する。

2. 自律運用管理システムとその課題

自律運用管理システムは現在人手で行われている企業システムの運用管理をシステム自身が持つ機能によって自動化するシステムである。

自律運用管理システムの実現には、管理対象の状態を把握するための資源情報管理機能が特に重要な役割を果たす。問題が発生したときに正確な対処を実行するためには、資源情報管理機能によって管理対象の状態を正確に把握しなければならない。例えば、サーバでディスクエラーなどの障害が発生した際に、サーバの詳細な構成情報を把握することができなければ、具体的な対処手順を決定することができない。正確な情報を把握するためには、資源に直接アクセスして状態を確認する方法が最も確実である。しかし、管理対象となる企業システムではネットワークなどによる遅延が無視できないため、複数回に渡って情報参照を繰り返す場合、管理対象の状態把握に長時間を要してしまう。これは、迅速な対処を阻害する要因となる。

情報収集による遅延を回避する手段として、資源情報管理機能内に情報をキャッシュして保持する方法がある。我々はポリシー制御による運用管理ミドルウェア Polimatica[2]の開発において MDS[3]によるキャッシュ機能を利用した。しかし、キャッシュした情報は必ずしも最新の資源状態を反映しているとは限らず、キャッシュした情報が古いために誤った自律制御を実行してしまう問題があった。例えば、CPU 利用率の増加に合わせて負荷分散制御を自律的に実行する場合、負荷分散の効果がキャッシュの情報に反映されなければ、繰り返し負荷分散制御が実行されてしまう問題がある。

情報収集の遅延を回避し、かつ、可能な限り最新の資源情報を提供可能な資源情報管理機能が求められる。

3. 情報更新スケジュール機能

提案する情報更新スケジュール機能は、キャッシュした情報を自律運用管理の管理ステップに応じてスケジュール更新する機能である。この機能は図1に示すスケジュール生成機能とスケジュール管理機能、及びキャッシュ更新機能で構成される。

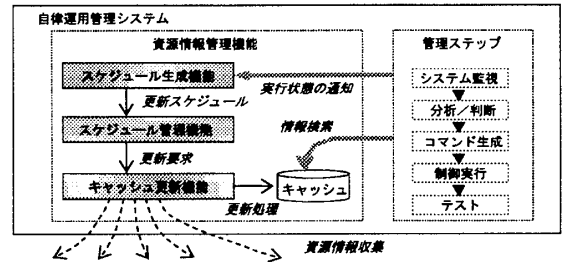


図1 情報更新スケジュール機能

自律運用管理システムで参照する全ての資源情報はキャッシュに格納され、キャッシュ更新機能によってキャッシュデータは随時更新される。いつ、どの情報を更新するかを定めたデータを更新スケジュールと呼び、スケジュール管理機能がこのデータを管理する。図2は host00、host01などの資源情報を、現在から何秒後に更新するか示したスケジュールデータの例である。このスケジュールデータを自律運用管理の管理ステップやシステムの負荷状況などに応じて生成する機能がスケジュール生成機能である。

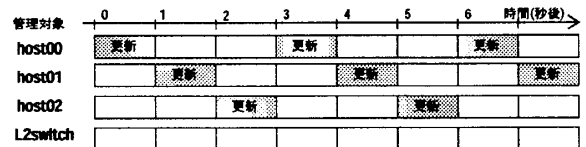


図2 キャッシュ更新スケジュールの例

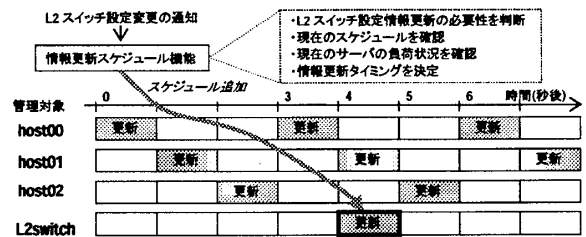


図3 スケジュール書き換えの例

[†] 日本電気株式会社 インターネットシステム研究所

自律運用管理の管理ステップは主に図1に示す、システム監視、分析/判断、コマンド生成、制御実行、テストからなり、各ステップで参照する資源情報はそれぞれ異なる。スケジュール生成機能では、資源情報の更新頻度や、各管理ステップで要求される資源情報を予め設定情報として用意し、管理ステップが遷移する際に設定情報に基づいて情報更新スケジュールを書き換える。例えば、制御実行ステップにおいて、L2スイッチのVLAN設定を変更した場合、キャッシュ上のL2スイッチの設定情報も更新する必要がある。図3はL2スイッチの設定変更処理後に更新スケジュールを追加する動作の例を示している。

4. 実装と評価

提案した情報更新スケジュール機能を実装して評価した。図4は評価システムの構成を示している。

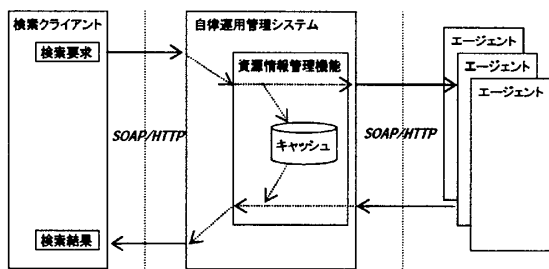


図4 評価システムの構成

自律運用管理システム内の資源情報管理機能を評価するため、資源情報の検索を要求する検索クライアントを用意した。検索クライアントはSOAP/HTTPで自律運用管理システムに検索要求を送る。資源情報管理機能は資源情報を提供するエージェントにSOAP/HTTPでアクセスして情報を収集する。提案方式では収集した情報をキャッシュして保持する。

サーバやネットワーク機器の情報を管理する15のエージェントを用意し、検索クライアントから全ての資源情報を参照する検索処理を実行した。このとき、処理の結果となる全資源情報を表すデータは約100KBであった。キャッシュを利用しない場合とキャッシュを利用した場合について、10回の検索処理の平均応答時間を比較した結果が表1である。測定にはPentium 4 3GHz、メモリ 1GBのLinuxマシンを用いた。

表1 検索応答時間の測定結果

	キャッシュ未使用	キャッシュ利用
資源情報管理機能内処理 (msec)	11840	68
資源情報管理機能外処理 (msec)	2328	2075
検索応答時間 (msec)	14169	2143

資源情報管理機能内処理の時間が応答時間に大きく影響を与える。検索時に情報を収集する場合、資源情報を取得するためのスレッドが同時に起動し、その全ての結果が得られるまで資源情報管理機能内での処理が終わらない。SOAP/HTTPによるオーバーヘッドと情報取得処理の待ち合わせが性能低下の要因となる。対して、キャッシュを用

いた場合は検索時に情報を取得するための処理が発生しないため、オーバーヘッドや待ち合わせの問題が生じない。

情報をキャッシュした場合は参照した情報が最新とは限らない点に注意しなければならない。検索によってクライアントが取得した各資源情報データの経過時間をプロットした結果を図5に示す。

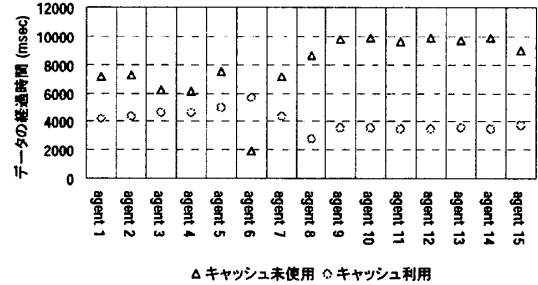


図5 参照した情報の情報鮮度測定結果

キャッシュを利用しない場合よりも、キャッシュを用いて提案方式を採用した場合の方が全体的に経過時間は小さく、新しい情報が得られている。

キャッシュを利用しない場合は、エージェントから最新のデータを取得できるが、検索処理全体にかかる時間に依存してクライアントがデータを取得するまでの経過時間は長くなる。一方、キャッシュを利用した場合、データの経過時間はキャッシュを更新した時刻に依存し、情報更新スケジュールによって各資源情報データの経過時間を制御することができる。評価に利用した情報更新スケジュールはキャッシュを利用しない場合よりも鮮度の高い情報を提供することを可能とし、監視ステップなどで有効に活用できる。

5. まとめ

本稿では資源情報参照の性能と資源情報データの鮮度を向上させる資源情報キャッシュ更新機能を提案した。キャッシュを用いたシステムの場合、鮮度の低下が課題となるが、提案機能により、キャッシュした情報の鮮度を情報更新スケジュールによって制御することが可能となり、自律運用管理システムに最適な資源情報管理システムを実現できる。

参考文献

- [1] J. O. Kephart and D. M. Chess. The vision of autonomic computing. IEEE Computer, 36(1):41--50, January 2003.
- [2] Y. Maeno, M. Kawato, S. Nishimura, F. Machida, and T. Kamachi. Polimatica: Abstraction for Customizable Private Virtual Organizations in Global Grids, The 2004 IEEE International Conference on Web Services (ICWS2004), July, 6-9, 2004.
- [3] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, Grid Information Services for Distributed Resource Sharing, In Tenth IEEE International Symposium on High Performance Distributed Computing (HPDC10), IEEE Press, August 2001.