

暫定対処を考慮した IT システム障害対処プラン自動生成システムの検討

永井崇之[†] 鈴木克典[†] 村瀬香緒里[†] 江丸裕教[†]

(株) 日立製作所 研究開発グループ[†]

1.はじめに

近年、企業内データセンタにおいてサーバ・ストレージといった IT インフラの仮想化技術が普及し、システム統合による設備コストの抑制や、機器台数の削減による消費電力削減といった恩恵をもたらしている。一方で、仮想化によりサーバやストレージが集約されたことで、障害発生時の影響範囲が拡大するという事態が起きている。さらに仮想化技術を利用することで、多様なベンダのハード・ソフトがデータセンタ内に混在することとなり、IT システムが複雑化している。このような状況では、障害が発生した際にシステム運用管理者が迅速に障害復旧作業を行えないことが多い。

この問題を解決するために、障害復旧作業を支援する研究が行われている。例えば、障害復旧作業に要する運用管理作業のうち、復旧作業に着目し、今後発生が予測される障害事象を管理者が入力すると、それを回避するための対処プラン(回避策)を生成する回避策推論システムが提案されている[1]。

本研究では、IT システムにおいて障害が発生し、複数の機器に影響が波及している場合に、障害を解消するための対策案を提示する障害対処プラン生成システムを、ストレージ装置の性能障害をユースケースとして具体化するとともに、その課題および解決方法を検討する。

2. 障害対処プラン生成システム

2.1.基本フロー

本研究で提案する障害対処プラン生成システムの処理フローを図1に示す。

提案システムでは、IT システムを構成する機器を監視し、異常状態となった場合は障害イベントとして管理者に通知する。なお、監視対象機器の性能異常に関しては、注意域に達した場合の閾値(第一閾値)と、業務停止が起き得る警告域に達した場合の閾値(第二閾値)の二段階で性能値を監視し、第一閾値超過時にイベント化することが一般的である。管理者は障害イベント群の中から原因となるイベントを特定し、対処プラン導出を指示することでプラン生成処理を開始する。(図中(1))

次に提案システムは、指定された障害イベントを解消する効果のあるプランを複数生成する。その際、発生しそうな障害イベント毎に事前にロジック化しておいた対処プラン生成方法と、IT システムの構成情報を格納した

CMDB(Configuration Management Database)を参照することでプラン生成を行う。(図中(2))

最後に、提案システムは事前に定義された評価指標情報や CMDB を参照し、各対処プランの確実性、所要時間、プラン実行により影響を受ける他機器の範囲および影響の大きさなどの特徴を分析し、定量化した上で、管理者に提示する。(図中(3)(4))その結果、管理者は提示された効果や特徴を見ながら要件にあう対処プランを選択することができる。

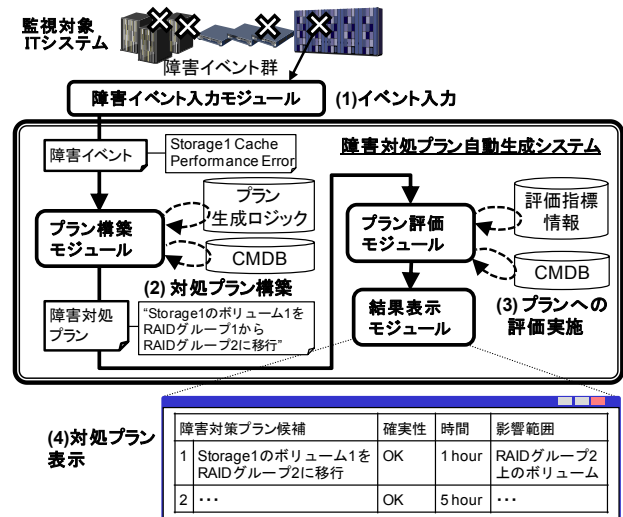


図1: 障害対処プラン生成システムの処理フロー

2.2.原因推論の必要性

提案システムにおいては、管理者が障害の根本原因となる障害イベントを特定した上で、対処プランの立案を実施する。しかし管理者は、障害の発生状況に基づき手動で原因を推論しているため、障害原因を正確には特定できない場合がある。管理者によって適切な障害原因が選択されない場合、発生している障害事象に対し適切な対処プランを提示できない恐れがある。

この問題に対処するため、IT システムにおける障害イベントの発生状況から真の障害原因を推論した上で、真の障害原因を解消するための障害対処プランを生成するシステムが提案されている[2]。本研究でも、管理者が障害原因として特定したイベントを引き起こしている別の原因がないかを分析した上で、真の障害原因に対し有効なプランのみを提示する。

以下、図2に示すストレージ装置におけるキャッシュのボトルネックを例に説明する。ストレージ装置は、複数の物理 HDD からなる RAID グループを持ち、RAID グループを論理的に分割したボリュームにホストコンピュー

A Configuration Planning System Countered for Failures on IT System Considering Temporary Action

[†] Takayuki Nagai, Katsunori Suzuki, Kaori Murase, Hironori Emaru, Hitachi, Ltd., Research & Development Group

タから書き込まれたデータを記録する。その際、ホストから受け取ったデータをキャッシュ上に一時的に蓄積する。データをキャッシュに書き込んだ時点で書き込み完了として次の処理に移行することで、ホストからの書き込み処理を高速化している。また、ストレージのキャッシュの内部は論理的に分割され、ホスト上で稼動する業務および対応するボリューム毎に使用する領域を分けることで、業務間の性能独立を図っている。

キャッシュ上の書き込み待ちデータの蓄積度合いを示すライトペンディング率(WP 率)という値が高まると、ホストからの書き込みを受け入れきれなくなり I/O 性能の低下が起きる[3]。キャッシュ WP 率が上昇する原因事象としては、以下の2つが考えられる。

原因 1 キャッシュへの書き込み量に比べ業務毎に割り当てられたキャッシュの量が少なく、書き込み待ちデータを蓄積するためのキャッシュの容量が不足。

原因 2 RAID グループの過負荷により、キャッシュから RAID グループ への書き込み処理が遅延し、キャッシュ上に書き込み待ちデータが多く発生。

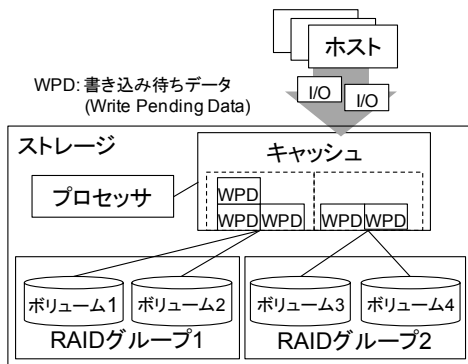


図 2: ストレージ装置のアーキテクチャ

提案システムでは、ストレージ装置で実際に起きている事象が前記原因のどれに該当するかを見極める。具体的には、キャッシュだけでなく接続する RAID グループでも障害イベントが発生している場合は原因 2、キャッシュのみでイベントが発生している場合は原因 1 が正しいとみなす。

さらに、上記見極め結果を踏まえ、各ケースにおいて有効な対処プランのみを提示する。原因1への対処としては、負荷の低い業務向けのキャッシュ領域から一部領域を切り出し、ボトルネックとなるキャッシュの容量を増やすことが有効である。原因 2 への対処としては、負荷の高い RAID グループ上のボリュームを、別の RAID グループに移行して負荷を下げるのが有効である。この結果、真に有効な対処のみを管理者に提示できる。

3. 暫定対処の提案

障害原因に対処するためのプラン実行に時間を要すると、プラン実行中に更なる性能の悪化を招く恐れがある。このため対処プラン立案の際は、原因に対処するプランだけでなく、暫定的な対処のためのプランも立案できる必要がある。

障害に対する対処の中には、対処開始から効果が発

揮され始めるまで時間が掛かるものが存在する。2.2.で述べた対処の中では、ボリューム移行が該当する。ストレージ装置のコピー機能では一般的に、旧ボリュームから新ボリュームのコピー作業が終わるまで旧ボリュームを利用し続けることとなるため、コピー終了後初めて効果を発揮するためである。一方、キャッシュ容量変更は、データをボリュームに書き出すことで空となった領域を随時用途変更するため、対処実行直後から効果がある。

以下、効果発現までに時間が掛かる対処を「遅延系対処」、対処実行直後から効果を発揮する手段を「即応系対処」と呼ぶ。遅延系対処からなるプランでは、設定変更完了に時間を要する場合、プラン実行中に更なる性能の悪化を招く恐れがある。

そこで提案システムでは図 3 に示すように、プラン開始後対策が完了するまでの時間 T1 と、トレンド予測の結果性能値が第二閾値に到達するまでの時間 T2 を予測する。遅延系対処を実行するプランで、かつ T1 が T2 を上回る場合、プラン実行途中で第二閾値を超え、業務停止に至る可能性が高い。

そのため提案システムでは、根本対策となる構成変更を実行する前に、一時的にホストからの書き込み負荷を下げる暫定対処を行うプランを提示する。この暫定対処に適した手段として、ホストの発行する書き込み量の上限をストレージ装置側で一時的に制限する事により、ボリュームへの負荷を低下させる方法が挙げられる。

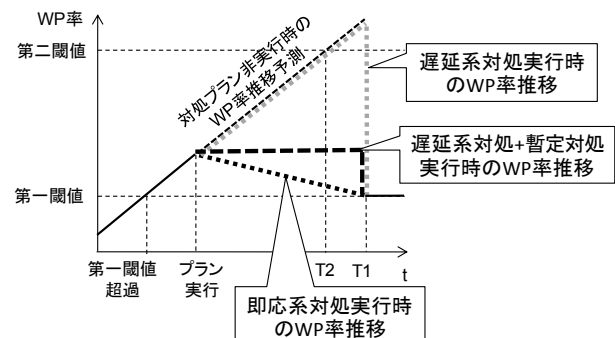


図 3: キャッシュ WP 率の推移モデル

4. まとめ

本研究では、対処プラン立案の際、原因に対処するプランだけでなく、状況の悪化による IT システム上の業務の停止を防ぐための暫定的な対処を加味したプランを立案するシステムを提案した。今後は、本方式の有効性を詳細に検証するため、評価を実施する。

参考文献

[1] 加藤裕, 敷田幹文: 障害予測における最適な障害回避手段の提示法, 情報処理学会インターネットと運用技術シンポジウム 2012 論文集, pp.110-116 (2012).
 [2] 永井崇之他: IT システム向け障害対処プラン自動生成システムの検討, 電子情報通信学会技術研究報告情報通信マネジメント 112, pp.125-130(2013).
 [3] (株)日立製作所: Hitachi IoT Platform Magazine「ストレージ I/O 性能の低下を防ぐ」<<http://www.hitachi.co.jp/products/it/it-pf/mag/susume/01/p02.html>> (2015).