3.1　フェールソフトに関する教訓（T1）

教訓 T1　サービスの継続を優先するシステムにおいては、疑わしき構成要素を積極的にシステムから切り離せ（“フェールソフト”の考え方）

問題

サービスの継続を優先するデータの非同期送受信（メッセージ交換型）のオンラインシステムで、サーバの自動切替えが失敗し二つのノードが稼働する事象が発生した。

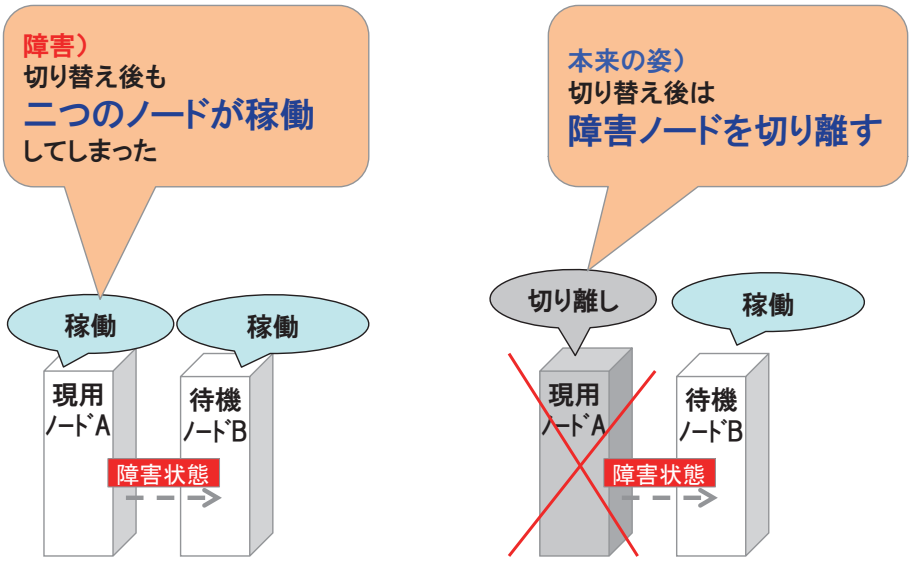


図 3.1－1　障害時のイメージ

原因

直接の原因はミドルウェア・OS の潜在バグである。根本原因は、それに対応してノード A 自身が自動停止しなかったことによるものである。

本システムはキューを使用したシステムであり、システム障害時にトランザクションが不完全に完了してもデータの整合性を保つように設計されている。このシステムにおいてノード A がノード B に『障害状態との誤ったメッセージ』を送り、ノード B も A の代わりに『稼働可能ステータス』に向けた遷移を開始した。そのため、ノード A も B も稼働状態となってシステム全体のステータスに不整合が発生した。

対策

応急対策としては、手動により障害となったノードＡを強制的に切り離した。

恒久対策としては、以下のようにフェールソフトの考え方を適用する。フェールソフトとは、システムの障害時の際にも、正常な部分だけで稼働を継続させることを重視した考え方である。

- **具体的な考え方**

  業務内容に基づいて、システムごとにポリシーを作成した上で、フェールソフトを適用する。ハードウェア機器の故障、ソフトウェアの処理プロセスの異常等があった場合には、その部位を積極的に停止させることでシステムから切り離す、場合によってはその系全体を放棄するといった考え方のもとに処理・対応する。

  一方、そのような状況下で一部の部位や系をシステムから切り離しても、システム全体としてのサービスは継続できるように、フェールソフトの考え方に基づいて設計・運用する。

- **この考え方に立つ理由**

  機器やソフトウェアそれぞれの動作継続を優先し過ぎてしまうと、予期せぬ障害の場合にサービスの影響がかえって大きくなってしまう場合があり、サービスの継続を優先させるためには、むしろ積極的に関連する部分をシステムから切り離す方が良い場合が多い。

  具体的には、ハードウェア、ミドルウェア、ソフトウェアにおいて、それぞれの機器やプロセスが自己診断等を基に個々に切替え・停止を判断する場合が多いが、予期せぬ障害においては、その動作が不安定となりコントロールできない状態になってしまうことがある。そのような場合には、周囲の機器やプロセスが、コントロールできない状態となっているおそれのある部位を強制的に切替え・停止させシステムから切り離すことにより、円滑に対応を行える仕組みを構築することで、サービス継続がより実現しやすくなる。

効果

フェールソフトを実装しておくことで、重大障害が発生しても業務が継続可能となり、取引の停止や報道発表につながるような重大トラブルが未然に防止できるようになった。

また、未知の障害が発生しても、システムを停止することなく業務の継続が可能となった。

教訓

サービスの継続を優先するシステムにおいては、疑わしき構成要素を積極的にシステムから切り離せ

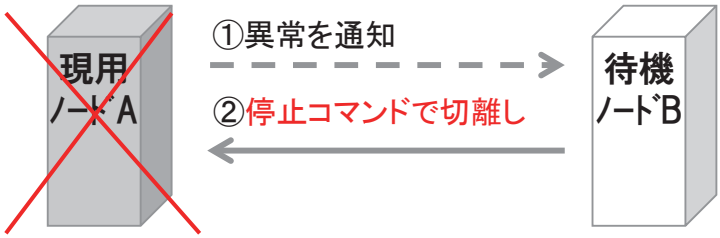
（サービスの継続と原因究明とのどちらを優先するかは、サービスの特徴に基づいてマネジメント層が決定すべきことである。上記対策は、比較的短いメッセージの１ラウンドの送受信によって完結するような、単純な処理において適用可能であり、適用に際しては十分な技術検討が必要である）。

【補足説明】対策の詳細

図 3.1－2 のようにフェールソフトの考えを取り入れ実装する。

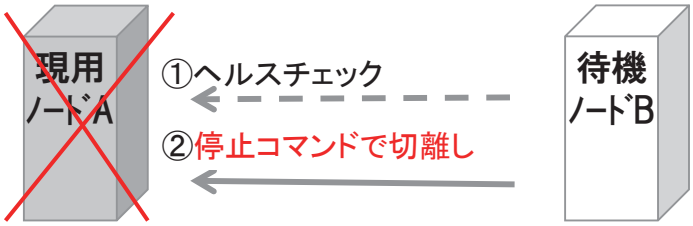
(1) 自身のヘルスチェックの場合

構成要素自身がヘルスチェックを行う場合、異常を検知して他系の構成要素にその旨の通知を行ったとき、通知を受けた系の構成要素は、念のため、通知を行った系の構成要素に対して停止コマンドを送る。



(2) 他系のヘルスチェックの場合

相互に他系の構成要素のヘルスチェックを行う場合、他系の構成要素の異常を検知して自身が現用系として動作するとき、検知した系の構成要素は、念のため、異常と判断した他系の構成要素に対して停止コマンドを送る。



(3) 自動停止できない場合は手動による停止で切り離し

自動停止できない場合のために、手動による停止を容易にするための仕組みを組み込んでおく。

図 3.1－2　フェールソフトによる切替え時のイメージ