監視の技 2　対応不要なイベントを抑止して障害コールを削減したい［ログフィルタリング・イベント集約］

現状の課題

システム障害が発生した際、ログは障害検知、原因特定の重要な手掛かりとなる。OS のログやアプリケーションのログを監視し、障害検知条件にマッチした時、障害の発生を知らせる「障害イベント」を発行する。しかし条件の設定によっては、障害とはいえないものを検知して障害イベントを発行してしまう。

例えば「ERROR が含まれる」「WARNING が含まれる」という単純な文字列を条件に指定した場合、大量の障害イベントが発行されてしまうことがある。また、一つの障害に起因して副次的な障害が発生する場合も同様である。これでは、大量に発行される障害イベントの確認作業に大きな負荷が掛かるだけでなく、重要なイベントを見落とすリスクも生じる。

OS やアプリケーションによって監視すべき範囲が異なるため、ログごとに障害検知の時間や通知方法などを細かく定義する必要がある。例えば、アプリケーションのメンテナンス中の障害検知は不要だが、OS リブート中に発生したハードウエア障害は基盤に影響を与える恐れがあるため通知する必要がある。こうした条件を洗い出して反映させることが求められる。

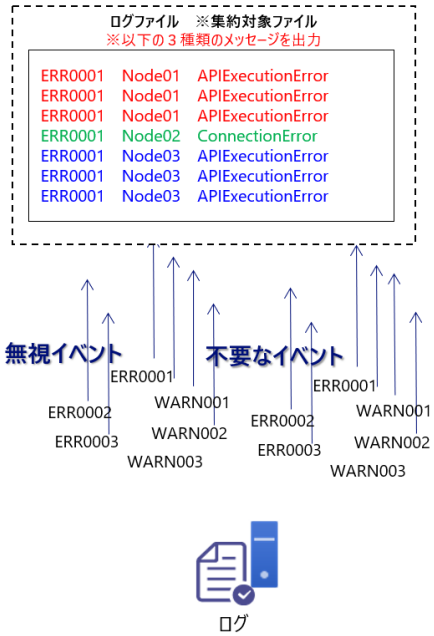
また、障害発生時、復旧までの間に同一の障害ログが連続で出力され続けてしまうケースでは、それぞれのログを検知して大量の障害イベントが発行されてしまう可能性もある。

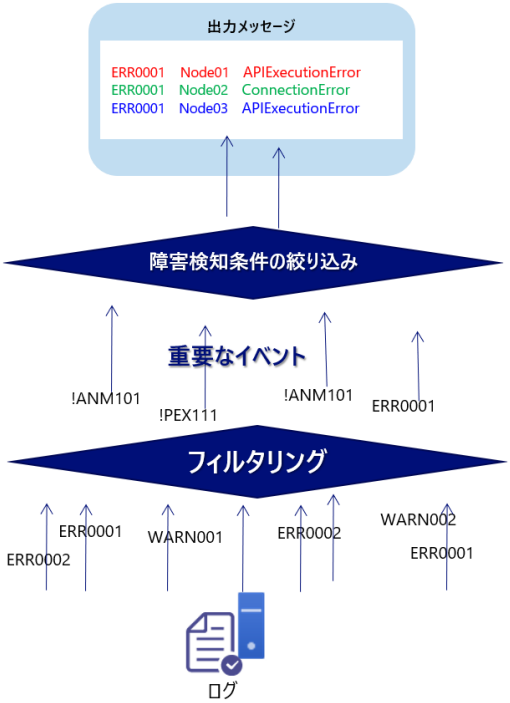
解決策

重要なイベントを見落とさないためには、まず不要な障害イベントを発行しないことから始めるべきだ。障害検知の条件を単純な文字列のみではなく正規表現で指定する。その他、一度に大量の障害を検知した時、根本原因となるものに対してのみ障害イベントを発行する。ここが一番のポイントになる。システムの連結テストや総合テスト時のログにどのようなものが出力されるのかを洗い出し、本当に対応が必要なものを障害検知の条件として定め、一方で対応が不要なものを抑止することで、本番環境で障害が起こった時の確認作業を効率化できる。

異なる監視範囲については、アプリケーションのメンテナンス中は監視を停止し、作業終了後に再開するよう監視時間の設定をしておく。OS リブート中はログの監視（＝障害検知）機能も停止するが、起動時にはリブート前に監視していた位置までさかのぼって障害有無を確認する必要がある。ログの監視時間や監視場所も設定条件の大事なポイントになる。

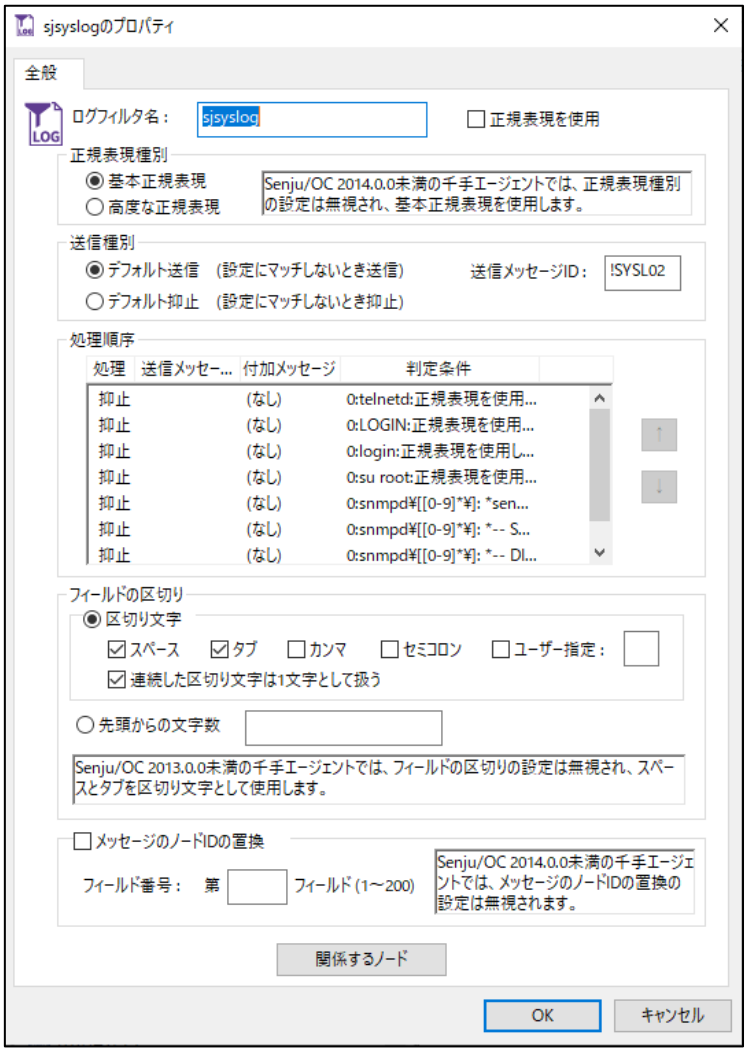
同一のログが連続で出力され続けるケースにおいては、一定期間内のログの検知を集約し、障害イベント発行を一度にして減らすことが必要となる。





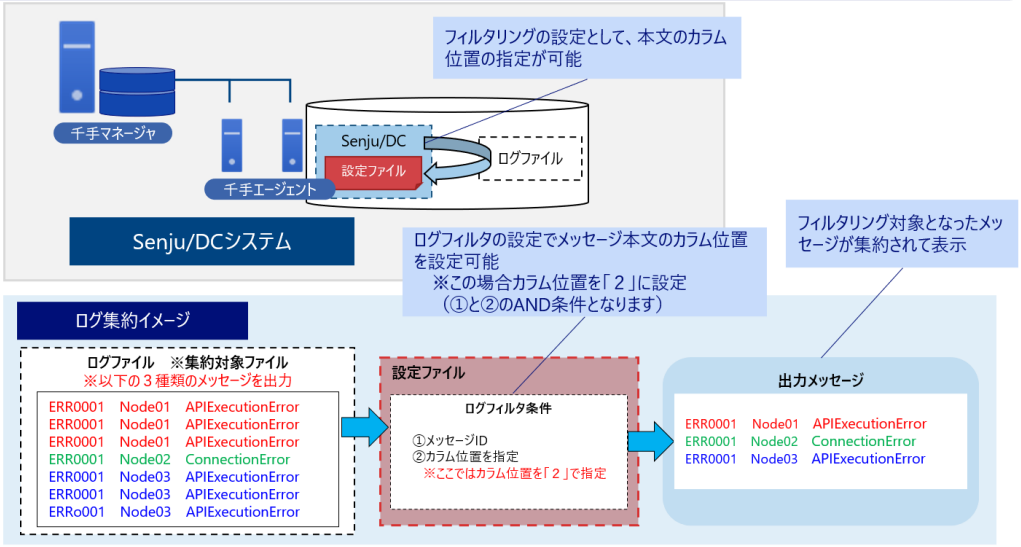
Senju Family での実践方法

Senju DevOperation Conductor(Senju/DC)ではイベントの発行を抑止する「ログフィルタ」機能を提供している。正規表現を指定できる他、検査対象を AND 条件で判定するなど、一つのログフィルタ内で複数の定義が行え、その条件をどのようにフィルタリングで使用するかの指定も可能だ。ログフィルタに掛かったものを別ファイルとして保存し、そのファイルのログを分析することで、発行するイベントを数件に絞ることもできる。キーとなる障害が通知された後、実際に障害対応する時になってその他の情報を参照するだけで事足りる。



上記に加えて「ログフィルタ候補」機能を活用することで、監視対象のファイルに対してあらかじめ複数のフィルタ設定を用意しておき、時間帯に応じてそれらの設定を自動で切り替えることができる。これにより、定期的な作業時など特定の時間だけ検知を抑止する等の柔軟な監視が可能となる。

ログ監視の停止・再開については、「ログ監視の位置記憶」機能を実装している。監視再開の際には前回ログフィルタを実施した最後の位置から、またはログの一番後からといった OS やアプリケーションの性質に合わせた監視設定を可能とする。

同一のログが連続で出力され続ける場合、「ログ監視による大量メッセージ出力の抑止」機能を活用することで、障害イベントを削減可能である。事前にログ監視機能の拡張設定ファイルに、メッセージ ID やノード ID、特定のカラムの文字列が同一である場合などの条件を記載することで、同一監視間隔内に発生したログに対して、該当の障害イベントを 1 つにまとめることができる。

より細かな条件設定によるフィルタリングでアラートを削減。
メッセージIDに加え、カラム位置でのアラート重複判定により、精度の高い集約が可能