ジョブ管理の技 24

障害が発生したサーバーのジョブを全てスキップして切り離し作業をしたい

現状の課題

用途（業務）によっては、複数サーバーに分散してジョブを稼働させるケースがある。その場合、分散しているジョブを順序よく時間内に完了させるのがジョブ管理ツールの主な役目である。複数サーバーで動くジョブは、複雑な先行・後続関係がある。システムが複雑化し、大規模化すると、それぞれのジョブの関係も複雑になり、障害時の対応も大変になる。

例えば、ノード障害が発生したサーバーで稼働するジョブが異常終了した場合、後続の別サーバーで稼働するジョブの実行を遅延させないために、異常終了したジョブは一旦スキップ終了させ、後続を先に動作させるというケースがある。その場合、異常終了したジョブをスキップし、後続が稼働しても、そのさらに後続に（ノード障害が発生したサーバーで実行する）ジョブがあると、またそのジョブが異常終了してしまう。

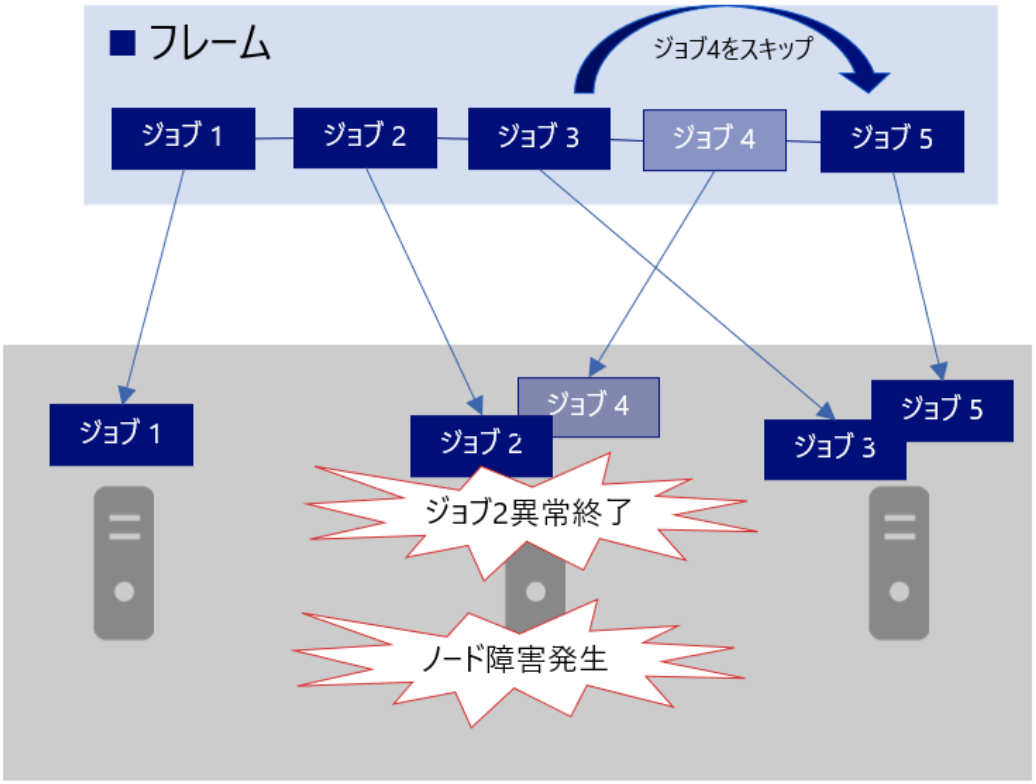
ジョブが異常終了する都度、ノード障害が発生したサーバーで動くジョブかを確認し、スキップを行うというオペレーションが必要になる。これが、障害対応が遅れる要因となり得る。

解決策

ノード障害が発生したサーバーで稼働するジョブを一括してスキップし、切り離しができれば迅速な障害対応が可能となる。具体的には、指定したサーバー（障害が発生したサーバー）で、この後稼働する予定のジョブの一覧を抽出し、そのジョブ全てをスキップ指定できればよい。毎回手動で作業を行うとタイムラグが発生し、ミスの危険性もあるので、他のイベント（CPU 使用率増やメモリー利用率増、など）をトリガーにして自動実行させるべきである。

Senju Family での実践方法

Senju DevOperation Conductor ではサーバー指定での「ジョブスキップコマンド」機能を提供している。コマンドとして提供しているので、メッセージアクション、ランブックと組み合わせ、サーバーのリソース監視（CPU やメモリーなど）で異常が発生したら、ジョブスキップコマンドを自動実行するといった連携もできる。


■ フレーム
ジョブ4をスキップ
ジョブ 1
ジョブ 2
ジョブ 3
ジョブ 4
ジョブ 5
ジョブ 1
ジョブ 4
ジョブ 2
ジョブ 5
ジョブ 3
ジョブ2異常終了
ノード障害発生